NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

ENHANCED RADIO FREQUENCY (RF) COLLECTION WITH DISTRIBUTED WIRELESS SENSOR NETWORKS

by

Mickey S. Batson

June 2007

Dissertation Supervisor: John McEachen

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2007	**3. REPORT TYPE AND DATES COVERED** Dissertation	
4. TITLE AND SUBTITLE: Enhanced Radio Frequency (RF) Collection With Distributed Wireless Sensor Networks		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Mickey Batson			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

In this research, a novel approach for conducting signals intelligence from a distributed network of wireless nodes is developed. The primary objective of this research is enhancing signal collection in a specified target direction. Two conflicting priorities are addressed. One is the time required to determine the target direction and form the beams. The other is the energy consumption involved in developing these solutions.

Two competing enhanced collection methodologies (ECM), ECM-1 and ECM-2, were developed and analyzed. ECM-1 uses a combination of time difference of arrival (TDOA) and adaptive beamforming. ECM-2 uses adaptive beamforming that performs a beamscan similar to phased-array radars. Additionally, two competing methods for forming the beams are developed. Method One uses data exclusively from the same elements. Method Two uses data from a new subset of sensors, for each iteration.

Analytical expressions were derived for energy consumption and the time required to develop, to compare the competing methodologies. ECM-1 is shown to be far superior to ECM-2 in both energy consumption and the time required to enhance signal collection, whereas Method Two is shown to be far superior to Method One in formation of the beams.

14. SUBJECT TERMS Sensor Networks, Adaptive Beamforming, Direction Finding, Time difference of Arrival (TDOA), Cross-ambiguity Function (CAF)			**15. NUMBER OF PAGES** 297
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ENHANCED RADIO FREQUENCY (RF) COLLECTION WITH DISTRIBUTED WIRELESS SENSOR NETWORKS

Mickey S. Batson
Commander, United States Navy
B.S. Electrical Engineering, University of Mississippi, 1988
M.S. Electrical Engineering, Naval Postgraduate School, 1998

Submitted in partial fulfillment of therequirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2007

Author: _____
Mickey S. Batson

Approved by:

_____	_____
John McEachen Associate Professor of Electrical. & Computer Engineering Dissertation Committee Chair	Murali Tummala Professor of Electrical & Computer Engineering
_____	_____
Weilian Su Asst. Professor of Electrical & Computer Engineering	Frank Kragh Asst. Professor of Electrical & Computer Engineering
_____	
Chris Eagle Lecturer	

Approved by: _____
Jeffery B. Knorr, Chair, Department of Electrical & Computer Engineering

Approved by: _____
Julie Filizetti, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this research, a novel approach for conducting signals intelligence from a distributed network of wireless nodes is developed. The primary objective of this research is enhancing signal collection in a specified target direction. Two conflicting priorities are addressed. One is the time required to determine the target direction and form the beams. The other is the energy consumption involved in developing these solutions.

Two competing enhanced collection methodologies (ECM), ECM-1 and ECM-2, were developed and analyzed. ECM-1 uses a combination of time difference of arrival (TDOA) and adaptive beamforming. ECM-2 uses adaptive beamforming that performs a beamscan similar to phased-array radars. Additionally, two competing methods for forming the beams are developed. Method One uses data exclusively from the same elements. Method Two uses data from a new subset of sensors, for each iteration.

Analytical expressions were derived for energy consumption and the time required to develop, to compare the competing methodologies. ECM-1 is shown to be far superior to ECM-2 in both energy consumption and the time required to enhance signal collection, whereas Method Two is shown to be far superior to Method One in formation of the beams.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my parents for giving me an insatiable desire to learn. I would like to thank my wonderful children for their thrills and inspiration. Most of all I would like to thank my beautiful wife for her support and untiring devotion.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

In this research, a novel approach for conducting signals intelligence from a distributed network of wireless nodes was developed. The primary objective of this research is enhancing collection in a specified target direction. The SIGINT/IW wireless sensor network consists of a large number of small sensor nodes that are distributed over a large operational area, to acquire information about signals of interest. These sensor nodes collaborate among themselves to form an ad-hoc network and disseminate the collected target information to a centralized controller node, e.g., UAV, UUV, etc. In this research, it is assumed that the clustering functions to create a sensor grid have been completed. It is assumed that perfect node location has been determined and that these data are available to the centralized controller node.

The centralized controller node, henceforth referred to as the central controller, acts as a reference antenna by gaining initial intercept and frequency of interest determination. The central controller is also tasked with maintaining frequency, phase and data synchronization among the nodes. Since it is envisioned that the sensors within this network will utilize low cost, omnidirectional antennas, adaptive beamforming is used in order to enhance the RF signal collection and/or information operations capability.

The task of conducting RF collection from a network of this type was divided into four steps. The first step is the determination of the frequency of interest of the target. The second step is the synchronization of the individual nodes. The third step is the determination of the direction of the target. Finally, the fourth step is the formation of the beam in the target direction. Two of these steps are specifically addressed in this research, steps three and four. The first step will be assumed to be accomplished by an implementation of the periodogram. The second step will be assumed to be accomplished through a “brute-force” method similar to the one presented by, Jenn, *et al*. [19] for wireless radar applications.

When addressing steps three and four, two conflicting priorities were addressed. One is the time required to determine the target direction and the time required to form the beams. The other is the energy consumption involved in developing these solutions. Two competing enhanced collection methodologies (ECM), called ECM-1 and ECM-2, were developed and researched. ECM-1 uses a combination of time difference of arrival (TDOA) and adaptive beamforming. ECM-2 uses adaptive beamforming that performs a beamscan similar to phased-array radars.

Additionally, two competing methods for forming the beams were created, Method One and Method Two. Method One created a subset of sensors from a randomly distributed antenna array. This method uses data exclusively from these elements and increases the number of iterations until the desired beam is formed. Method Two creates a new subset of sensors, from a randomly distributed array, for each iteration, until the desired beam is formed.

Analytical expressions were derived that proved that Method Two provides better power management across the sensor network. Additionally, Method Two was able to mitigate the grating lobes that are present in arrays where the antenna element spacing is greater than a half-wavelength.

ECM-1 is described as follows. The central controller will utilize the cross-ambiguity function to calculate the TDOAs between the individual sensors and the central controller. The central controller will then use the Newton-Raphson technique (NRT) to determine a line of bearing to the target emitter. The central controller using Method Two will begin the beamforming process. The central controller randomly selects the number of elements required to create the appropriate beam for a given azimuth and elevation of a target emitter.

ECM-2 is described as follows. The central controller will then coordinate the direction-of-arrival (DOA) determination using beamscanning. For this method, the central controller employs Method Two for creating the beams. The central controller randomly selects the number of array elements needed to form the desired beam. This

process is repeated until a 360 degree scan has been completed. The central controller will then make a determination as to the direction of the highest received power.

Analytical expressions, for energy consumption and time to develop a solution, were derived to compare ECM-1 to ECM-2. ECM-1 was shown to be far superior to ECM-2 in both energy consumption and the time required to formulate a solution.

In this research, a novel idea for conducting signals collection from a wireless sensor network was developed. To facilitate signals collection from this network, an enhanced collection methodology, ECM-1, was developed. ECM-1 is comprised of a unique rapid, energy efficient technique for determining the direction of the target and a unique adaptive beamforming technique, Method Two. Analytical expressions were derived to compare ECM-1 with ECM-2. Several simulations were developed and implemented to study the differences between ECM-1 and ECM-2.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In this research, a novel idea for conducting signals collection from a wireless sensor network was developed. Our current national intelligence resources are limited in responsiveness and number. Current tactical intelligence collection sensors lack capability, flexibility and are limited by the inability to connect with national sensors. Both Joint Vision 2020 and Sea Power recognize the role of intelligence collection networks in full-spectrum dominance and the need for persistent intelligence, surveillance and reconnaissance operations [20, 23]. Advances in software radio, wireless communications and electronics have enabled the development of smart, disposable microsensors that can be deployed in the battlespace for conducting Signals Intelligence (SIGINT) and Information Warfare (IW) [2, 73]. These improvements in sensor network technologies continue to decrease the size, weight and costs of sensors and increase their resolution and utility [9, 65].

Previously, wireless sensors focused on the collection of data with low bandwidth requirements. The methodology in this research will require and assume that sufficient bandwidth is available to transport the required information back to a centralized node for processing. This research will leverage and expand on current research areas: "opportunistic array" research by Jenn, *et al.* [20], a second is adaptive beamforming for sensor networks by Tummala, McEachen, Vincent, *et al.* [68], and another is time difference of arrival estimation by Loomis, *et al.* [29].

The "opportunistic array" concept is an integrated digital phased array radar, where the array elements connect wirelessly and are placed randomly over the surface of a ship. The unique requirement of having these elements function coherently led Jenn, *et al.* [21]. to design individual transmit and receive (T/R) modules. Specifically, these researchers focused on the wireless local oscillator (LO) distribution and control methods, microstrip patch antenna design, commercial-off-the-shelf (COTS) hardware investigation and synchronization.

In a separate research project at the Naval Postgraduate School, Vincent, *et al* explored the concept of beamforming in order to minimize power requirements for data

exfiltration. The results were positive analytically proving that by treating a sensor grid as an array of randomly distributed antenna elements, a beamforming algorithm could be used to enhance the transmission range of the network.

Building and testing a rapidly deployable adaptive SIGINT/IW sensor grid is different in application, but similar in approach to the wireless radar approach. Likewise, using beamforming techniques for reception or conducting offensive information operations of target signals of interest (SOIs) is different but similar in approach to beamforming for the purposes of data exfiltration. In this research, a novel approach for conducting signals intelligence from a distributed network of wireless nodes was developed.

The primary objective of this research is enhancing RF collection in a specified target direction from a SIGINT/IW wireless sensor network. The proposed sensor network consists of a large number of small sensor nodes, K, distributed over an area, A_1, to acquire information about signals of interest. These sensor nodes collaborate among themselves to form an ad-hoc network and disseminate the collected target information to a centralized controller node, e.g., UAV, UUV, etc. In this research, it is assumed that the clustering functions to create a sensor grid have been completed. It is assumed that perfect node location has been determined and that these data are available to the centralized controller node.

To begin, this research explored the fundamental tenets of conducting SIGINT/IW. In order to enhance RF signal collection from this sensor network, these tenets were divided into broad categories. First is the determination of the frequency of interest of the target. Second is the synchronization of the individual nodes. Third is the determination of the direction of the target. Fourth, is the formation of the beam in the target direction. Two of these areas are specifically addressed in this research, categories three and four. The first category will be assumed to be accomplished by an implementation of the periodogram. The second category will be assumed to be accomplished through a "brute-force" method similar to the one presented by Jenn, *et al*. [21] for the wireless radar application.

To address categories three and four, two conflicting priorities were addressed. One is the time required to determine the target direction plus the time required to form the beams. The other is the energy consumption involved in developing these solutions. Two competing enhanced collection methodologies (ECM) for direction-finding within this wireless antenna array network were developed and analyzed. ECM-1 uses a combination of time difference of arrival (TDOA) and adaptive beamforming. ECM-2 uses adaptive beamforming that performs a beamscan similar to phased-array radars.

Analytical expressions for energy consumption and time to develop a solution, were derived to show that ECM-1 was far superior to that of ECM-2. ECM-1 was shown to be far superior to ECM-2 in both energy consumption and the time required to formulate a solution.

In order to better focus on category four, two competing methods for forming the beams were developed. Method One selects a subset of sensors from a randomly distributed antenna array. This method uses data exclusively from these elements and increases the number of iterations until the desired beam is formed. Method Two creates a new subset of sensors, from a randomly distributed array, for each iteration, until the desired beam is formed.

Analytical expressions were derived that proved that Method Two provides better power management across the sensor network. Additionally, Method Two was able to mitigate the grating lobes that are present in arrays where the antenna element spacing is greater than one half-wavelength.

This dissertation is organized as follows. Chapter II will provide an overview of wireless sensor networks and lay the foundation for the SIGINT/IW wireless sensor system, to include characteristics and the proposed architecture. The chapter will also address some of the challenges in realizing this network. Chapter III will summarize previous work. The discussion will include digital array radar sensors, basic array antennas, communications architecture, wireless local oscillator element synchronization, and beamforming. Chapter IV will discuss an adaptive SIGINT/IW sensor network. The chapter will include a discussion on antenna theory and array properties, spectral estimation techniques and element synchronization. Chapter V will be a discussion on

subspace methods used for direction of arrival determination. Chapter VI will discuss the direction finding techniques of time difference of arrival and beamforming. Chapter VII will discuss the enhanced collection methodology developed in this research, and Chapter VIII will provide conclusions and discussions on future work.

II. OVERVIEW OF WIRELESS SENSOR NETWORKS

As discussed in [2] and [68], wireless sensor networks consist of devices that combine the functionality of sensing, computation and communication into a single device capable of self-organization and inter-device connectivity. Wireless sensor networks can be used in a number of military and civilian applications. As seen in Figure 1, these sensor nodes can be deployed quickly and left unattended by humans, which make them very attractive for military applications [2, 68]. Once deployed, these SIGINT/IW sensor nodes have the ability to acquire information about SOIs and forward the information back to a centralized node for processing and relaying back to a regional National Security Agency/Central Security Service (NSA/CSS) facility, e.g., NSA/CSS Hawaii or a tactical collection platform.

The chapter is organized as follows. Section A will discuss the proposed SIGINT/IW sensor network architecture. Section B will provide an overview of SIGINT/IW sensor network characteristics. Finally, Section C lays out some of the perceived challenges in realizing this SIGINT/IW sensor network.

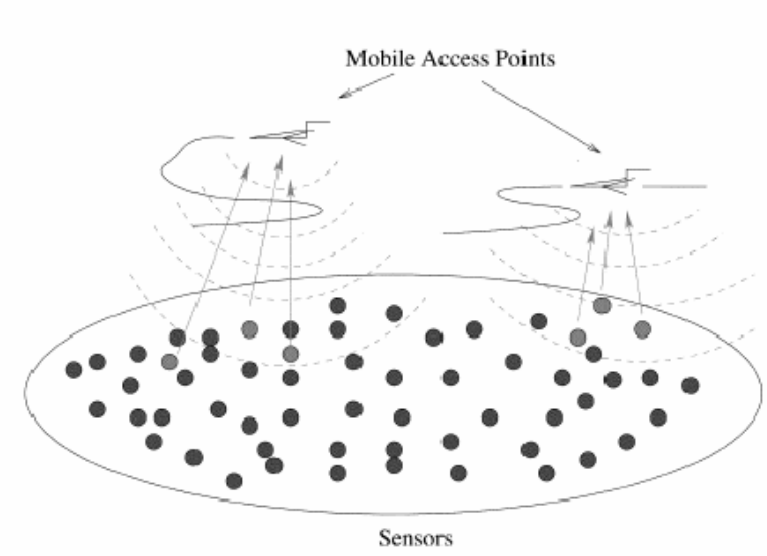


Figure 1. Random deployment of sensor nodes by an unmanned aerial vehicle (UAV). [From Ref. 2]

A. SIGINT/IW SENSOR NETWORK ARCHITECTURE

Several wireless sensor network architectures have been proposed in the literature. Figure 2 shows a general wireless sensor network architecture, which consists of a sensor field, a sink node and a satellite through which the user can access the sensor field [2]. A two-tiered hierarchical clustering sensor network architecture is the one envisioned in this research. A central controller node (i.e., UAV, UUV, etc.) provides the link between the satellite and the sensor nodes, and each sensor node has the ability to route the data to the central controller node. The central controller node in turn sends the data to the user node via the satellite [73].

The central controller will be the reference antenna and is tasked with determining the frequency of the signal-of-interest, maintaining frequency, phase and data synchronization among the remaining nodes within the cluster. The network is comprised of K sensors spread over an area, A_1. As seen in Figure 2 the target emitter lies far outside the sensor field. This is in contrast to most traditional sensor networks that monitor events that traverse through the sensor field.

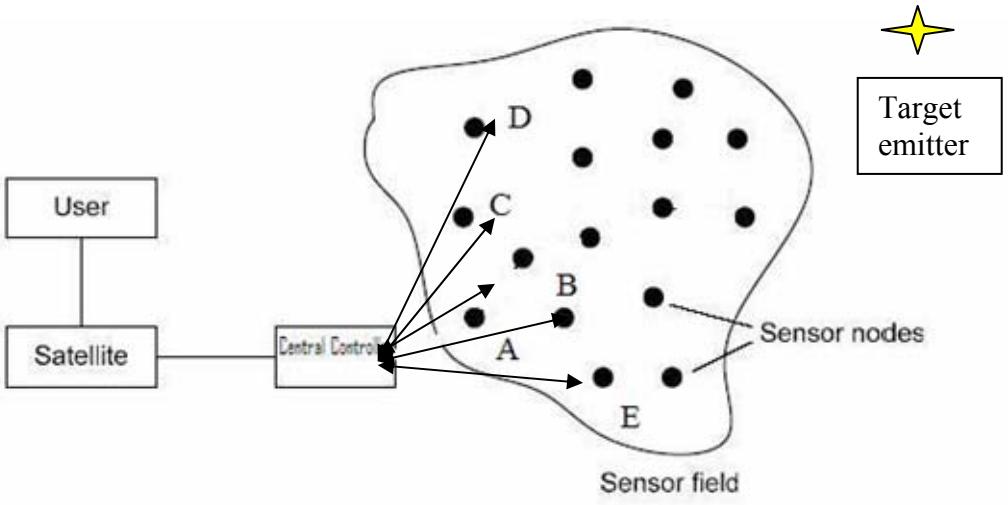


Figure 2. Proposed wireless SIGINT/IW sensor architecture. [After Ref. 2]

The requirements of sensor nodes dictate the need for an efficient system design. The sensors themselves will require much power, so minimizing the organizational

functions, and data exfiltration power requirements will be extremely important. This design criterion also governs the interconnection of sensors.

B. CHARACTERISTICS OF A SIGINT/IW SENSOR NETWORK

As stated before it is advantageous to separate the process of conducting SIGINT/IW into manageable sections. The SIGINT/IW process was divided into four broad categories. First is the determination of the frequency of interest of the target. Second is the synchronization of the individual nodes. Third is the determination of the direction of the target. Finally, is the formation of the beam in the target direction. Two of these areas are specifically addressed in this research, categories three and four. The first category will be assumed to be accomplished by an implementation of the Periodogram. The second category will be assumed to be accomplished through a "brute-force" method similar to the one presented by , Jenn, *et al*. for the wireless radar application.

The SIGINT/IW sensor network will be a collection of devices that once deployed can self-organize allowing for unattended operations. This self-organization will require the nodes to arrange themselves into a functioning network. The network will be centrally controlled. The nodes have the ability to collect radio frequency information and communicate this information to the central controller.

A SIGINT/IW sensor node will be assumed to possess the following distinguishing characteristics. These distinguishing characteristics will be the ability to demodulate and analog-to-digital convert intermediate frequency (IF) signals, as well as the capability to modulate and digital-to-analog convert IF for the purposes of conducting offensive information operations. The sensors will have to have the ability to store large amounts of data required for the SOI, direction finding, beamforming, etc [4, 57].

The storage capacity requirements will be much greater than those of typical sensor networks in order to account for channel access and processing delay at each sensor node, propagation delay between nodes, formulation of a given solution at the central processor and the ability of a third-party, i.e., an operator, to determine if sustained collection is required [17]. This type of sensor network must operate as power

efficiently as practical to extend battery life. It will be assumed in this research that the sensors possess the capability of obtaining power from the surrounding environment, e.g., solar, sea, etc [6, 27, 65].

C. NETWORK CHALLENGES

In order to realize the SIGINT/IW network, several network challenges will need to be addressed; these include, but are not limited to medium access control, localization, and synchronization techniques. Medium access control is the method through which the various nodes access the medium in order to relay data. Localization is the ability of a node to determine its physical location. Node synchronization is important in order to function as a coherent antenna array [6, 35].

1. Medium Access Control (MAC)

The communication functionality of a sensor network node follows a layered protocol architecture. Figure 3 describes the typical layered protocol stack of a sensor network node. As laid out in [65], the application layer provides mechanisms for analog-to-digital conversion. The network layer is responsible for seamless transfer of information, and the data link layer provides fair access and is responsible for error-free transmission. The physical layer provides a means of sending and receiving a bit stream [64].

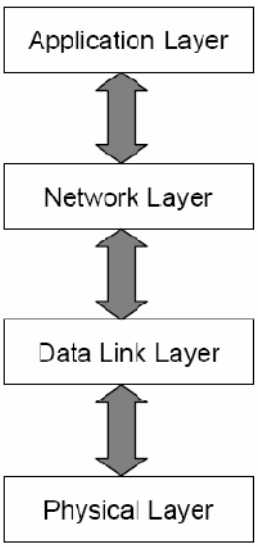


Figure 3. The protocol stack for a typical sensor network. [From Ref. 55]

Contention management is important in helping to avoid or recover from collisions, and minimize the overall penalty in terms of wasted network resources in shared-medium environments, such as contention-bus or wireless networks. Typically, these methods include random access, demand assignment, and fixed assignment [11].

Due to the expected high sampling rates of the individual SIGINT/IW sensor nodes and the perishiability of the data collected, choosing or designing a MAC suitable for this type of network will be paramount. Each type of MAC scheme has advantages and disadvantages, but it is envisioned that a fixed assignment MAC will provide the best service. This idea is based on the notional fact that the data would be arriving at each node simultaneously and all of the data is important. All of the data is required by the central node in order to form the appropriate solution.

2. Localization

One of the primary objectives of deploying a SIGINT/IW sensor network is to detect the presence of SOIs and identify the positions of their emitters. This, in turn, poses a requirement on the sensor nodes to determine their own positions, thus some method for location discovery is needed. This section provides an overview of the relevant position location techniques and methods. More complete reviews and algorithm comparisons have been presented in [28, 38, 41, 44, 61]. Sensor networks require the extension of existing methods by finding ways to use measurements such as range or angle between pairs of unknown-location nodes.

In a wireless environment, networked transceivers can obtain distance and angle measurements by one or a combination of the following ways:

1. A sensor node's location can be calculated directly and independently, if the sensor node is in range of multiple reference nodes. The sensors would calculate the estimated range by measuring the received signal strength (RSS) when a reference signal reaches the object [41]. The RSS can be measured as part of normal communications. Therefore, no additional energy or bandwidth is required. However, due to multipath reflections of the reference signal this technique can be unpredictable.

2. The sensor node's location can also be computed from the time-of-arrival. Time-of-arrival is simply the measure of the moment at which the signal first arrives at the receiver. The time delay between the sender and the receiver can be used to calculate

the angle of arrival or the distance when compared to a reference node This technique requires accurate measurement of the time-of-arrival of the line-of-sight (LOS) signal. This technique can also suffer from multipath and noise.

3. Finally, a sensor node may use the angle of arrival to calculate position. The angle of arrival (AOA) technique requires accurate angle information, typically using phased antenna arrays with multiple antennas of known separation to perform angular calculation [28].

Due to the extreme requirement for perfect sensor location, timing synchronization and covertness, each node will be assumed to have some form of precision location capability such as a GPS receiver. This capability will ensure all nodes are able to be localized with a minimum of communications between sensors. This eliminates the requirement for beacon signals, thereby reducing the adversary's ability to detect our network. It will be shown in Chapter III, Section G, subsection 2, how adaptive beamforming is affected by node location errors.

3. Synchronization

A persistent technological challenge arising from the development of the wireless opportunistic array concept has been the need to perform synchronization of the array elements to provide time and frequency references. Synchronization allows nodes the capability to determine their relative position when deployed randomly. To achieve data aggregation, the sensor must be able to precisely determine the instant in time at which an event occurred in order to recognize duplication [35, 65]. Previous research [14] has shown that control of the elements' phase is possible via a wireless LO signal. In [28] two synchronization techniques were studied for a dynamic environment where the transmission paths will be changing and unpredictable. Generally, the techniques aim to get phase corrections for synchronization without explicit measurement of the elements' position displacement. These two techniques will be discussed in more detail in Chapter III, Section E.

This chapter provided an overview of the wireless SIGINT/IW sensor network. The architecture for this network was discussed to lay the foundation for the rest of the dissertation. Some of the general characteristics of the basic functions and missions of the wireless sensor network were covered. Finally, some of the networking challenges, not addressed in this research, but nevertheless are critical to networking, were examined to

give an appreciation for the complexity of this network. Now, a discussion of previous research relevant to the wireless sensor network will be offered.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SUMMARY OF PREVIOUS WORK

This chapter summarizes some of the ongoing research relevant to the wireless SIGINT/IW sensor network. Sections A – E are comprised of wireless radar research conducted by Jenn, *et al*. [20] The focus in those research areas was on the sensor elements, networking and synchronization.

Section F recaps some of the time difference of arrival research conducted by Loomis, *et al*. [29] Finally, Section G will discuss the research into beamforming techniques conducted by Tummal, McEachen, Vincent, *et al*. [68]

A. DIGITAL ARRAY RADAR SENSORS

The opportunistic array research has led to the design of a digital transmit antenna with hard-wired array elements that operate from 2 to 2.5 GHz. The antenna was constructed using the Analog Devices AD8346EVAL quadrature modulator boards. Computer simulations using a genetic algorithm (GA), which has advantages for pattern formation when the array geometry is random or aperiodic, were used to verify formation of the radiation beam with randomly located elements. This part of the research demonstrated the viability of the transmit component of the phased array [28].

In addition, the bandwidth characteristics of the AD8346EVAL modulator board were investigated. Another commercial product, the Analog Devices AD8347EVAL quadrature demodulator board was configured to operate as a receive phase shifter. Reference [28, 40] investigated the design of the complementary phased array receiver architecture using the AD8347 demodulator. Different types of time-varying phase weights for a linear frequency modulated signal were demonstrated for the transmit side to improve the phase distortion and increase the operating bandwidth of the phased array [28].

While the previous research was focused on the creation of a wireless radar it is easy to see how the motivation is different but the technology is similar to the technology required for a SIGINT/IW sensor. For clarity the phrase SIGINT/IW has replaced the phrase wireless radar to demonstrate how these nodes have potential applications for use

in the wireless SIGINT/IW antenna network. A basic block diagram of the antenna array system architecture is shown in Figure 4. This block diagram is an adaptation of the one presented in [28]. For clarity, only a single wireless module with a central controller is shown. The central controller should be capable of communicating wirelessly with hundreds or even thousands of array elements that are self-standing SIGINT/IW modules.

For remote deployment applications, the central controller can be located in a UUV, UAV or a non-mobile central controller, while the array elements will be randomly distributed.

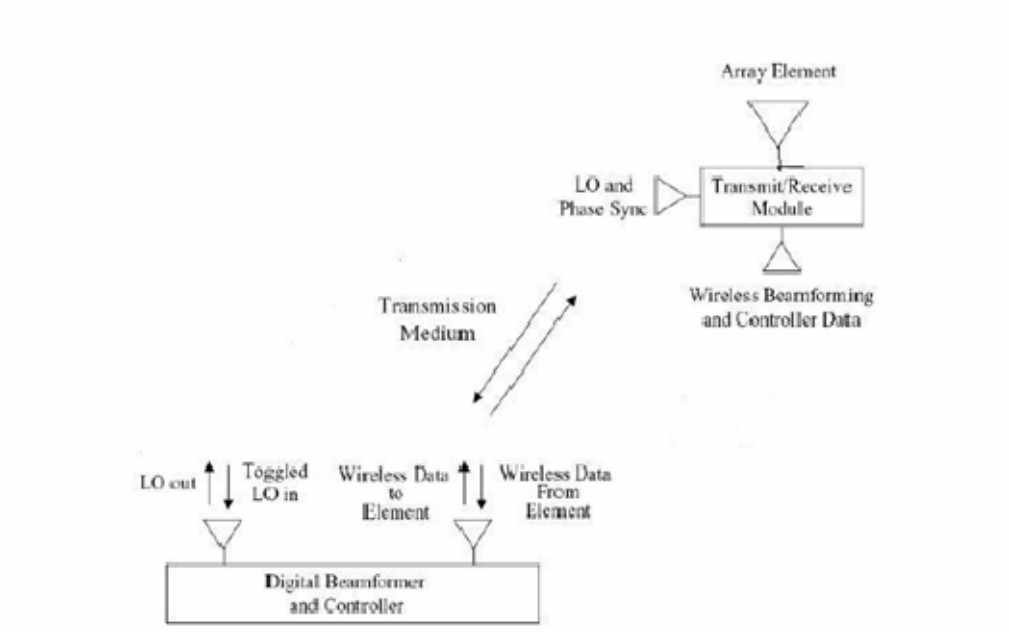


Figure 4. Network architecture for the SIGINT/IW. Only one sensor displayed. [After Ref. 20]

The general operational concept of the distributed array is as follows. At each array element, the received signal is downconverted to baseband after low-noise amplification, quantized (with the A/D) and the in-phase and quadrature data returned to the central controller for processing. Conversion directly to baseband will increase the sophistication of the sensor nodes, but it will drastically reduce the storage requirements at the nodes and it will reduce the network traffic, as well. The central controller computes the necessary control data – both phase and amplitude weights – for each element. In addition the central controller will maintain and deliver the appropriate signal

of interest (SOI) parameters. These are combined with the local oscillator (LO) and synchronization signals and are passed wirelessly to all array elements. A detailed architecture for each SIGINT/IW sensor is shown in Figures 5 – 6. If offensive information operations are desired, the digital baseband signal is generated by the direct digital synthesizer (DDS), converted to analog (with the D/A), directly up-converted to the operating band and power amplified. The LO reference signal is distributed wirelessly. An active phasing technique is used to compensate for element dynamic position changes and propagation channel changes.

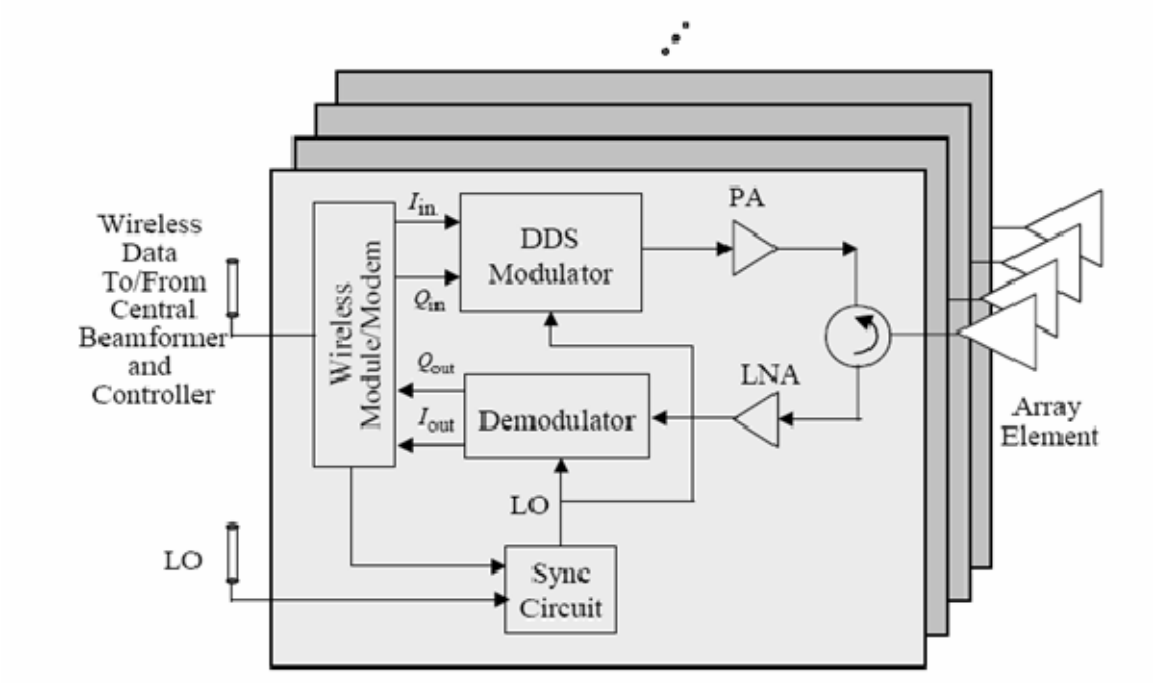


Figure 5. System architecture for the SIGINT/IW Tx/Rx module. [After Ref. 20]

The wireless opportunistic array concept demands very high data rate wireless communication. Numerous self-standing SIGINT/IW sensors continuously communicate element localization and synchronization signals, beam control data, and digitized SIGINT signals wirelessly with the central signal processor [28]. A block diagram of the prototype SIGINT/IW sensor is shown in Figure 6 [28].

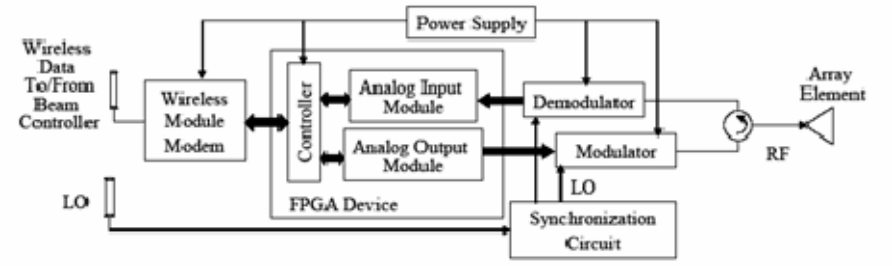


Figure 6. Block diagram of the demonstration SIGINT/IW sensors. [After Ref. 21]

B. BASIC ARRAY ANTENNA

Jenn, *et al.* [28] created and tested a low-profile, broad-band U-slot microstrip patch antenna. The antenna was designed to operate in the upper VHF/lower UHF frequencies. A set of simple design procedures was proposed to provide approximate rules that result in a good "first-pass" design with prescribed characteristics that require minimal tuning.

C. COMMUNICATION ARCHITECTURE

In order to provide localization, pass synchronization, beamscanning and beamforming coefficients, other targeting data, etc., a wireless connection was established to integrate the elements throughout the opportunistic array. A proposed, communication architecture for the wirelessly networked opportunistic digital array radar (WNODAR) was presented in [28] and is shown in Figure 7.

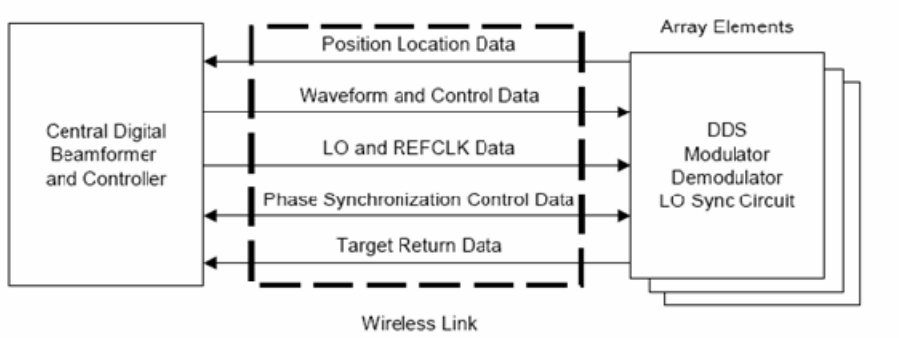


Figure 7. Communication architecture for the opportunistic array. [From Ref. 28]

The data flow required to conduct SIGINT/IW is analogous to if not the same as the data flow required to perform the functions of the WNODAR. Periodically each sensor will send its position data to the central digital beamformer and controller via the position location data signal [20]. In the SIGINT/IW network this function is contained within one system referred to as the central controller. When the central controller has determined the element locations it begins to calculate the appropriate digital amplitude and phase weights for each array element. The central controller then broadcasts this data via the waveform and control data signal to all array elements [28]. The local oscillator information is required by the modulator and demodulator. Reference clock data is required for the DDS signals. This information has been combined into a single waveform. A pulse train is transmitted from the centralized controller and the pulse train envelope detected and used for timing [28]. The carrier can be extracted and used for the local oscillator. Each Tx/Rx module will incorporate hardware for performing the synchronization. The phase synchronization control data is used to phase synchronize all the array elements [20]. The central controller will be used to establish a beamscanning pattern. When a signal is received, the signal is demodulated and the SOI data is sent to the central controller for processing. [20] If transmitting then the amplitude and phase corrected waveform is then modulated, amplified and transmitted. A summary of the wireless data determined in [20] is shown in Table 1 [28].

Description	From	Data Rate
Waveform control	Beam controller	2 bit/s
Synchronization control	Beam controller	1 bit/s for synchronization command 1 bit/s for phase correction command
Phase weights control	Beam controller	4 bit/s
Received radar signals	SIGINT/IW sensors	16 bit × 2 ×100 kS/s = 3,200,000 bit/s = 3.2 Mb/s

Table 1. Summary of wireless data requirements. [From Ref. 28]

In [28], contemporary wireless communication systems were used for the prototype WNODAR sensors. A updated survey of the findings derived from [28] and [65] are presented in Table 2. It gives a brief overview of some contemporary commercial consumer wireless communication systems. Current systems are not

sufficient to support both the high data rates and large numbers of nodes at the appropriate distances required for the SIGINT/IW Network [28, 45, 65].

System or Standard	Frequency	Data Rate	Standard Operational Range
802.11a	5 GHz	54 Mbps	100 m
802.11g	2.4 GHz		100 m
802.11n	2.4 GHz	200 Mbps (typical); 540 Mbps (max)	250 m (indoors)
WUSB (UWB)	3.1 to 10.6 GHz	400 Mbps	< 10 m
Bluetooth 3.0	6 – 9 GHz	480 Mbps	15 m
802.15.4 (Zigbee)	2.4 GHz and 915 MHz	250 Kbps (max)	70 – 300 m

Table 2. Performance of commercial consumer wireless communication systems. [After Ref. 28]

D. WIRELESS LOCAL OSCILLATOR (LO) SIGNAL DISTRIBUTION AND TRANSMISSION SYSTEM

As previously stated the functions of the WNODAR and the SIGINT/IW system are similar. One of the most fundamental and key components of both systems will be the ability to maintain synchronization between the central controller and the sensors. One of the key tenets of this synchronization process will be the distribution of the local oscillator (LO). In the WNODAR project a method of wirelessly distributing the LO was explored. The technique from [21] is described here to demonstrate a possible solution to the distribution of the LO that will be encountered in the realization of the wireless SIGINT/IW antenna network.

Distribution of the LO signal is required for coherent array operations. As shown in [20], a wireless LO was successfully demonstrated in a laboratory environment. A sinusoidal LO signal was transmitted, and then received by two AD 8346 modulator boards operating at 2.4 GHz. During this laboratory experiment it was also shown that control of the elements' phase is possible via a wireless LO signal.

Several critical aspects of the opportunistic array concept were demonstrated using COTS hardware in the 2.4 GHz band. Low cost, high-performance modulators and demodulators are available at this frequency.

The LO reference signal must be distributed to the modulators and demodulators. Synchronization of the LO reference signal is necessary to lower the sidelobe levels and improve the pointing accuracy of the beam [28]. The LO reference signal can be distributed wirelessly; a synchronization circuit is used to control phase corrections due to element dynamic position changes and propagation channel changes. In the next section a more thorough review will be presented on the two synchronization methods presented in [28] to synchronize the elements in a dynamic environment.

E. ELEMENT SYNCHRONIZATION

As previously mentioned, synchronization of array elements to a common reference is required for the SIGINT/IW network to function as a coherent antenna array. Control of the elements' phase is possible via a wireless LO signal, but in dynamic conditions the transmission paths will be changing and unpredictable. To overcome this, two different synchronization techniques were proposed in [28]. A brief overview of those techniques will be presented in this section.

Synchronization of array elements in time and frequency ensures that the emissions from all elements converge coherently on the SOI, increasing average power and signal-to-noise ratio (SNR). In each SIGINT/IW sensor, the use of a DDS, demodulator and modulator will require precise phase-synchronization of multiple synthesized RF output signals to one another for coherent detection and integration. For our application, frequency is synchronized by a common wireless LO signal, and therefore the key focus is to provide time or phase synchronization. In [28], two techniques to perform element synchronization for the adaptive antenna array were investigated.

The "brute force" synchronization technique [28] is a systematic adjustment of the array element phases. It is easy to implement with some hardware incorporated in each array element and in the central digital beamformer and controller. Figure 8 shows a

detailed diagram of the synchronization block that is required in each of the SIGINT/IW sensors. Each synchronization block comprises a modem and controller connected to a phase shifter and a switch. When the switch is positioned for synchronization operation (as shown) the LO signal is passed through a circulator, low-noise amplifier (LNA), phase shifter and then retransmitted back and compared to a reference signal at the central controller. Under normal operation (switch opposite as shown), the LO signal is sent out to the modulator and demodulator for coherent beamforming. [28]

At the start of the synchronizing cycle, the central controller sends out each element's address in turn. When the element is selected, the switch is selected to the synchronization position, the received LO signal is shifted by ϕ_n and sent back to the central controller. At the central controller, the LO signal from element *n* is compared with the received LO signal from the reference element.

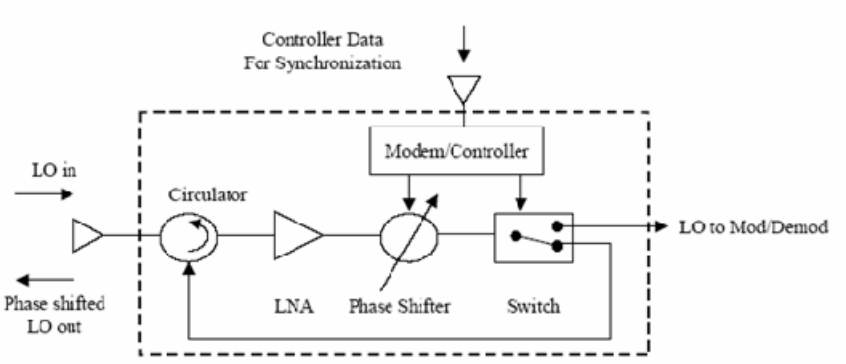


Figure 8. Diagram of a synchronization block for one element. [From Ref. 20]

Figure 9 shows the general concept to perform phase synchronization for element n. As described in [28], the LO signal from the central controller arrives at the synch block of each element at a different phase, $e^{-j\beta r_n}$, where β is the wave number and r_n is the distance from the controller to the element *n*. One element is selected as the reference element and it receives the corresponding LO signal $e^{-j\beta r_{ref}}$. As stated in [28], the objective is to synchronize all the elements to the reference element by adjusting the phase shifter ϕ_n to correct for the difference in path length $\left(r_{ref} - r_n\right)$.

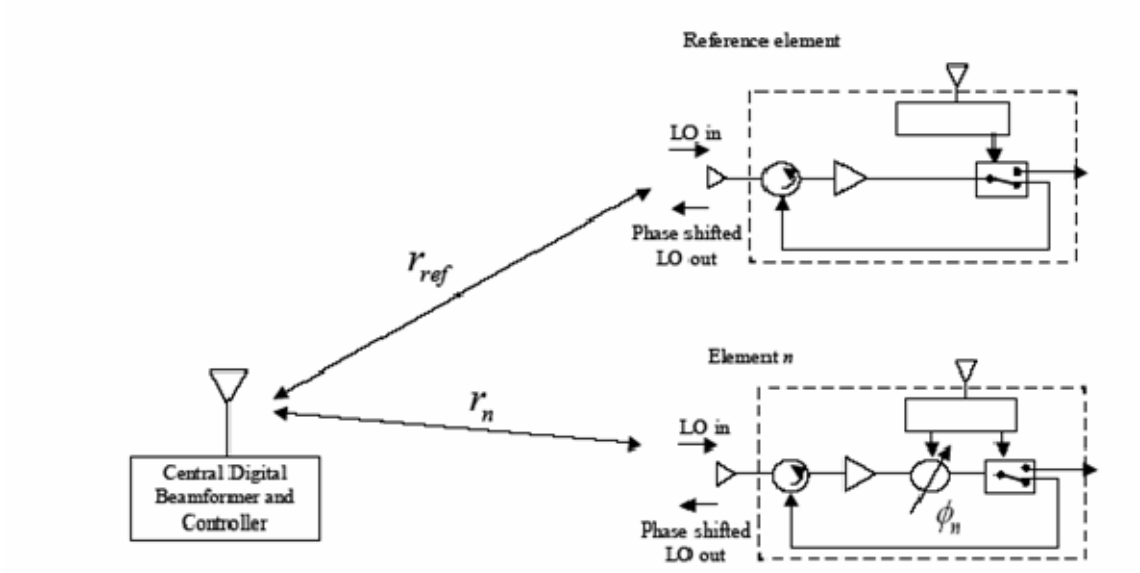


Figure 9. Phase synchronization using the "brute force" technique. [From Ref. 20]

Figure 10 shows the architecture for use with the beam tagging technique that was used in [28].

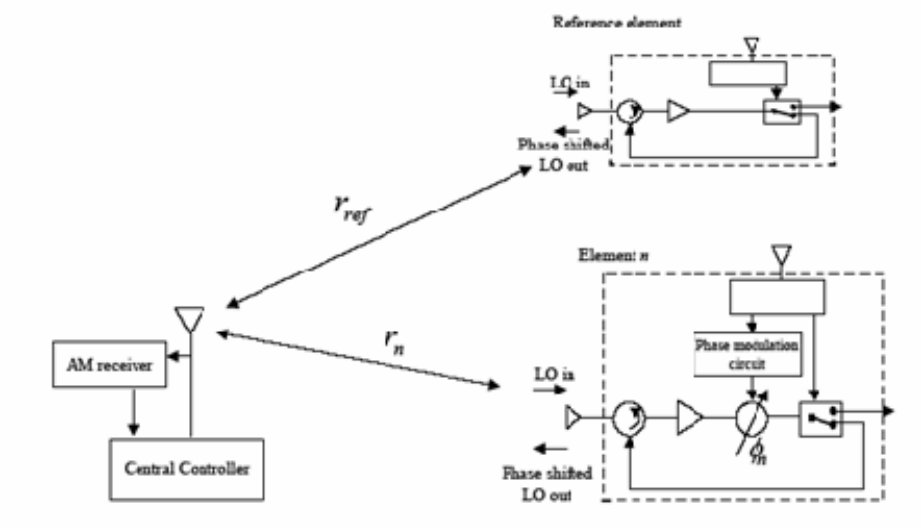


Figure 10. Phase synchronization using the "beam tagging" technique. [From Ref. 20]

The performance of these techniques were tested for use with the SIGINT/IW network. These results will be presented in Chapter IV, Section D.

F. TIME DIFFERENCE OF ARRIVAL (TDOA)

TDOA has been researched extensively by Loomis, *et al*. [29] The primary focus of this research has centered around the determination of time delay between high-speed mobile receivers that are maintaining the same relative distance to each other. This research will leverage this previous research specifically in the areas of calculating the actual time difference utilizing the cross-ambiguity function and determining a line of bearing based on the estimated time difference calculations using the Newton-Raphson Technique. This research will tailor these approaches to the specific characteristics of the SIGINT/IW network, specifically that the sensors are not moving.

G. BEAMFORMING

In [68], a three-tiered hierarchical clustered sensor network architecture was considered. A simulation model was developed and implemented in MATLAB® to study the application of beamforming using sensor clusters to establish a communication link with a UAV primarily utilizing a 2-D array. Different sensor node densities, position errors and sensor node failures were simulated to investigate their effects on the antenna beam generated by the sensor cluster [68].

The results were encouraging showing that the average sidelobe levels decreased while the maximum average power gain increased when the number of sensor nodes increased. In addition, the shape of the main lobe remained relatively unchanged when the number of sensor nodes changed.. However, it will be shown in this research that the mainlobe is highly frequency dependent. Results also showed that the presence of position errors and sensor node failures reduced the maximum average power gain of the antenna beam and increased its sidelobe levels, but the shape of the main lobe remained relatively unchanged [68].

Follow-on studies [73] confirmed consistency in the shape of the antenna main lobe under varying conditions. Therefore the beam can be steered towards the UAV as in the case studied in [73] in an adaptive manner as the UAV flies over the sensor field. This implies that the energy of the antenna beam can be directed towards the UAV, resulting in higher signal-to-noise ratio, and, hence higher channel capacity. Additionally, the

maximum average power gain of the main lobe can be increased by increasing the number of nodes in the sensor cluster, thereby increasing the transmission range between the sensor cluster and the UAV [73].

Previously, a discussion was presented on how the node density affects the array. Now a discussion will be offered on how node element failures and node location errors impact the array. Geolocation depends on the knowledge of the elements' position and this is even more crucial in digital beamforming. The individual elements are placed in open, available areas and the positions will be random. This fact must be taken into account in the signal processing to avoid degradation in the sidelobes, gain and beam pointing [28].

Now a presentation will be given, which analytically describes the effects of element failures and position errors.

1. Sensor Failures

Element failures will be a reality in sensor networks that will need to be mitigated. An analytic explanation articulated in [27, 68] will now be used. The fractional loss L_f in the main lobe gain for a two-dimensional MxN uniformly displaced array with equal weights due to element failures is given by [27, 68]

$$L_f = \left(\frac{M'N'}{MN} \right)^2, \tag{1}$$

where $M'N'$ is the number of operating elements and MN is the total number of elements. Since element failures are random, the operating probability of any element p_0 is given by

$$p_0 = \lim_{MN \to \infty} \left(\frac{M'N'}{MN} \right), \tag{2}$$

Hence, the expected value of the main lobe gain, relative to an error-free array, is reduced by a factor of p_0^2 [28]. Figure 11 shows the comparison between the performance a two-dimensional array with the elements separated by an equal distance and a two-dimensional array with elements uniformly distributed within a given area. The simulation used a sub-array of $MxN = 250$ nodes uniformly distributed within an area, $A_2 = 25m^2$.

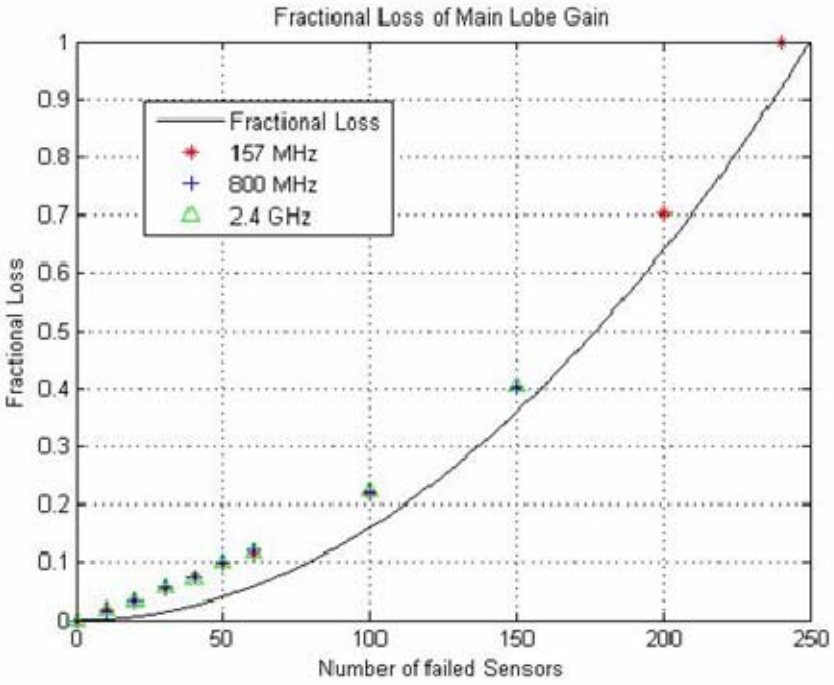


Figure 11. Fractional Loss in the gain of the mainbeam resulting from node element failures.

The simulated results were slightly poorer than that of the analytical prediction. However, the results did consistently follow the curve as seen in Figure 11. The results demonstrate how robust the beamforming algorithm of [73] is in the presence of node failures. The antenna array was able to sustain the loss of up to 40% of its elements before significant reduction in the gain of the mainbeam. What is interesting to note is that even though the mainbeam in all cases was reduced; the sidelobes were reduced as well. This allows the antenna array to maintain a direction finding capability even in these adverse conditions. In the next section a discussion on the effects of node location errors will be presented.

2. Sensor Location Uncertainty

The fractional loss L_p in the main lobe gain for a two-dimensional MxN array with a equal distance between elements and equal weights is given by [27, 73]

$$L_p = e^{-\sigma^2_{\Delta_p}}, \tag{3}$$

where $\sigma^2_{\Delta_p}$ is the variance of the phase error, assuming gaussian phase errors [73]. As per [57], the variance of the phase error is related to the variance of the node location error through the relationship shown in Equation (4).

$$\sigma^2_{\Delta_p} = \left(\frac{2\pi}{\lambda} \sigma_x \right)^2, \tag{4}$$

It can be seen in Figure 12 how the array with uniformly distributed elements within a given area using adaptive beamforming out performs the two-dimensional array that has equal spacing between elements.

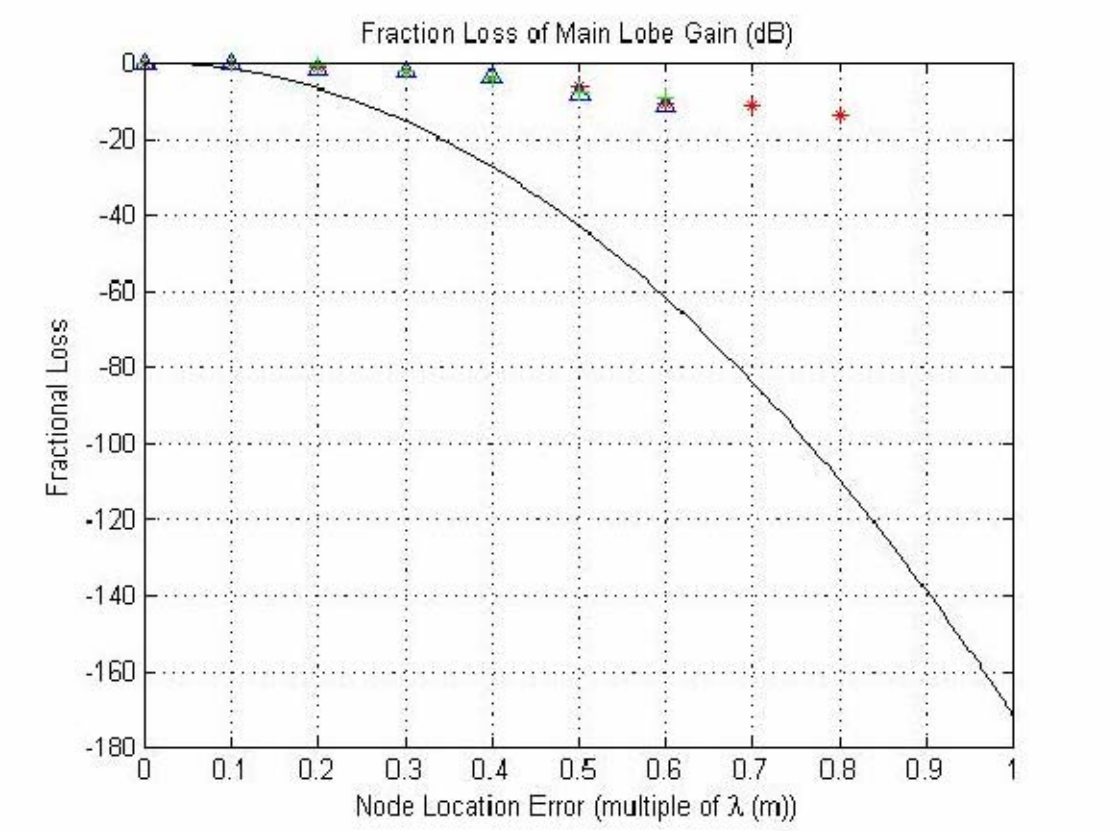


Figure 12. Fractional Loss resulting from node location errors.

Just like in the case of node element failures the fractional loss is not dependent on frequency. Figures 11 and 12 show the robustness of the LMS adaptive beamforming algorithm of [73]. This robustness is the exact properties desirable for applications within the SIGINT/IW network.

The research conducted in [68] was primarily focused on the exfiltration of data. The focus of adaptive beamforming in this research will be for the purposes of enhancing collection and offensive information operations utilizing a randomly distributed sensor array. Now a more thorough discussion will be presented on adaptive networks and direction finding.

This chapter summarized some of the previous work that has been conducted in areas germane to this dissertation. The chapter offered insight into research being conducted into the individual RF signal collection sensors, both the receivers and the associated antenna and an associated synchronization methodology. This chapter also provided discussions of research being conducted in the area of a proposed communications architecture. Finally, the chapter looked at time difference of arrival and Beamforming. This dissertation will use the research conducted in both of these areas to develop the proposed SIGINT/IW wireless antenna system and the enhanced collection methodology.

IV. ADAPTIVE SIGINT/IW SENSOR NETWORK

This chapter will provide an overview into some of the basics of conducting RF signals collection. First, a discussion on basic antenna theory and antenna array properties will be provided. The concept of spectral estimation will be discussed in order to provide the reader with a general understanding into how the proposed SIGINT/IW system will determine the frequency being transmtted from the target. Finally, a discussion will be offered on the synchronization of the individual sensor nodes.

A. ARRAY THEORY

It will be assumed that a single element will have an isotropic radiation pattern, which means the radiation is distributed equally in all directions. Therefore, it has low values of directivity and gain [3, 14]. Geolocation systems require very directive characteristics and high gain to meet the demand of high resolution. Two methods can be used to realize these requirements. As discussed in references [3, 14], the first method is to increase the electrical size of the antenna. The second is to form an array by assembling the elements in a specified pattern, but keeping the original size of the individual elements [3, 14].

Since the SIGINT/IW nodes are envisioned to be low power and minimally capable, the network needs to take advantage of the numerous elements present within the array. An antenna array can take advantage of the multi-elements and known separations between elements to combine the outputs and affect the antenna array pattern. This pattern will quite often differ from the pattern of the individual elements. An array has the ability to achieve the directional performance of larger antennas. By varying the phase and amplitude of the individual element outputs before combining, the overall array pattern can be steered in the desired user's direction without physically moving any of the individual elements [3, 14].

By the principle of pattern multiplication, the overall radiation pattern $F(\omega,\theta,\phi)$ is found as the product of the individual element radiation patterns $\xi(\omega,\theta,\phi)$ with the array factor $\Psi(\omega,\theta,\phi)$

$$F(\omega,\theta,\phi) = \xi(\omega,\theta,\phi)\Psi(\omega,\theta,\phi) \tag{5}$$

The array factor is determined by the positions of the elements relative to the central controller, as well as the phase and amplitude levels of the received signal. The array factor, and thus the overall pattern of the array, can be continuously scanned or adapted by adjusting the relative phase and amplitude levels between the elements [7, 14, 60].

A linear array is one in which the elements are aligned along a straight line. If all of the elements lie in a plane, the array is a planar array [7, 14, 60]. Examples include the linear array, a circular array, and arbitrarily shaped planar arrays. Due to its simplicity, the uniformly spaced linear array shown in Figure 13 will be used to show the phase relationship between the received signal at the different elements for a given angle of arrival ϕ_s measured along the main axis of the array. Since the signal present at element two has traveled a distance $d\cos(\phi_s)$ farther than the signal present at element one, its phase will lag behind that of element one by $\beta d\cos(\phi_s)$ where $\beta = 2\pi/\lambda$ is the phase propagation factor or wave number. This relation holds for a narrowband signal. A narrow band signal is defined as a signal whose modulating signal bandwidth is much less than the modulated signal [7, 14, 60].

The narrowband assumption allows us to assume that the only difference between the signal present at element two and element one is the phase shift induced by the extra distance traveled and is not significantly affected by the modulation during this time. An example in [60] demonstrates this concept. The complex envelope of a narrowband signal is shown in Equation (6) [7, 14, 60]

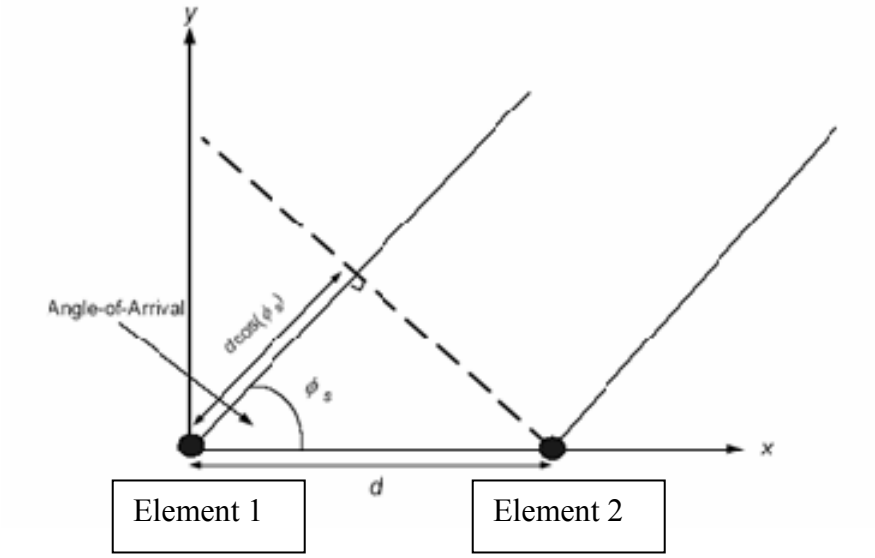


Figure 13. Two-Element Linear Array. [After Ref. 14]

$$s(t) = A(t)e^{-j(\omega_0 t+\gamma(t))} \tag{6}$$

where $A(t)$ is the signal amplitude, $\gamma(t)$ is an arbitrary phase shift and ω_0 is the center frequency. Assuming isotropic elements, $\xi(\omega_0)=1$, and taking the received signal at element one as the reference, the received signals $b_i(t)$ for a uniform linear array with element spacing d can be represented in matrix form as

$$\Psi(\phi_s)=\begin{bmatrix} 1 \\ e^{-j\beta d\cos(\phi_s)} \\ e^{-j\beta 2d\cos(\phi_s)} \\ \vdots \\ e^{-j\beta(N-1)d\cos(\phi_s)} \end{bmatrix}, \quad b(t)=\Psi(\phi_s)s(t) \tag{7}$$

where N is the number of elements, ϕ_s is the angle of arrival, and $\Psi(\phi_s)$ is the array factor. For simplicity, the frequency dependence of $\Psi(\omega,\theta,\phi)$ has been dropped for the narrowband case, and the elevation angle θ is assumed to be zero relative to the element's boresight. The array factor must be carefully measured to calibrate the array for direction finding experiments [7].

Now this concept will be extended to an $Mx1$ array with isotropic elements. Three areas in particular will be discussed. Two areas involve the baseline spacing, d, grating lobes and mutual coupling. The third area involves center frequency jitter, which can cause beam squinting.

1. Grating Lobes

The spacing between antenna elements is critical in the design of a phased array. This is because when the spacing exceeds certain critical values, grating lobes will occur [31]. Grating lobes are additional sidelobes that have the same amplitude as the main beam. It is normally undesirable to have grating lobes, and in order to avoid them, the element spacing is required to meet the condition given by

$$\frac{d}{\lambda_h} \leq \frac{1}{1+\left|\sin\phi_s\right|}, \tag{8}$$

where λ_h is the wavelength of the highest operating frequency and ϕ_s is the scan angle.

Based on this condition, when scanning to endfire at $\phi_s = \pm 90^\circ$ the first grating lobe will occur when the spacing is greater than $\lambda_h / 2$. For a comparison see Figures 14, 15 and 16. Figure 14 has $\lambda_h / 2$ spacing, Figure 15 has $3\lambda_h / 2$ and Figure 16 has $2\lambda_h$ [14].

It can be observed in Figures 14, 15 and 16, for the uniform linear spaced array used in the above calculations, that the antenna pattern is changed drastically as the spacing for the antenna is changed. This directly affects the amount of signal and noise that is collected and further analyzed. As the number of main lobes increases, the signal and noise, are no longer uncorrelated. As will be shown later too much correlation exists between the signal and noise for the classical subspace algorithms like MUSIC and Maximum Likelihood Method to separate them into individual components. Other signal processing methods will need to be employed to mitigate these grating lobes.

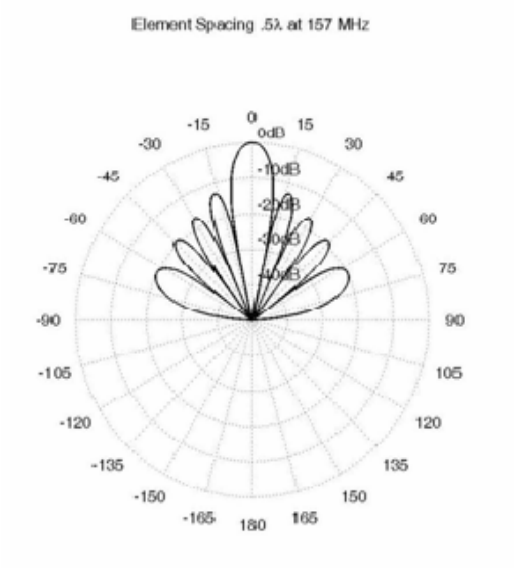


Figure 14. Array pattern for a 10 element Uniform Linear Array with half-wavelength spacing of the elements at a frequency of 157 MHz.

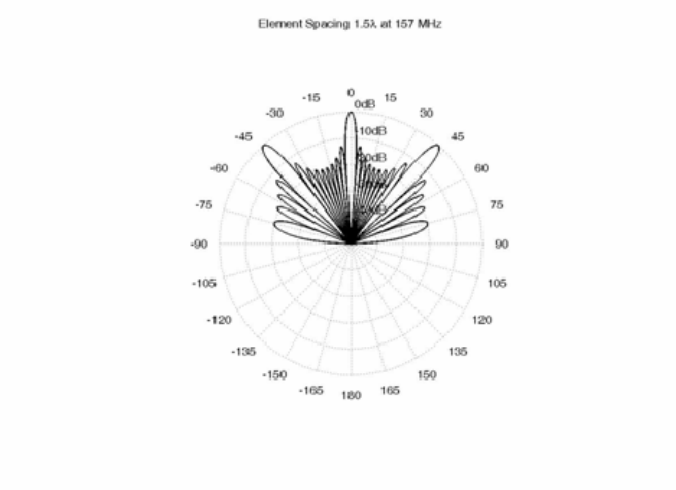


Figure 15. Antenna pattern for a 10 Element ULA with one and half-wavelength spacing at a frequency of 157 MHz.

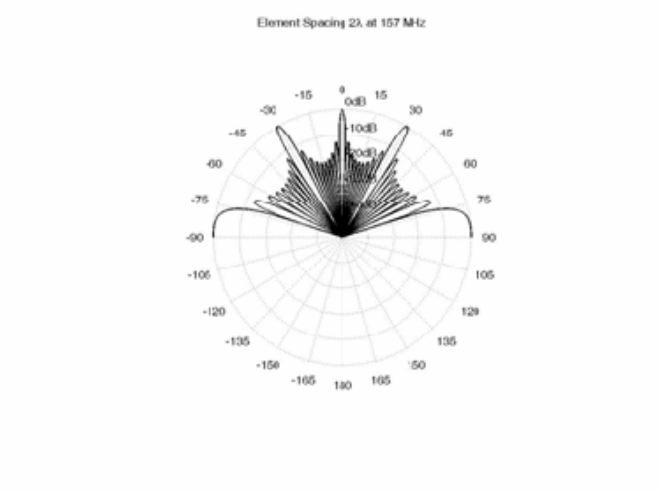


Figure 16. Antenna pattern for a 10 Element ULA with two times the wavelength spacing at a frequency of 157 MHz.

The element spacing used in the above simulations was not a realistic spacing given the random nature of deployment of our sensors. The sensors in the SIGINT/IW network on average would most likely be much further apart. For a more realistic spacing of one sensor every 10 meters one can see in Figure 17 that the situation is only getting worse.

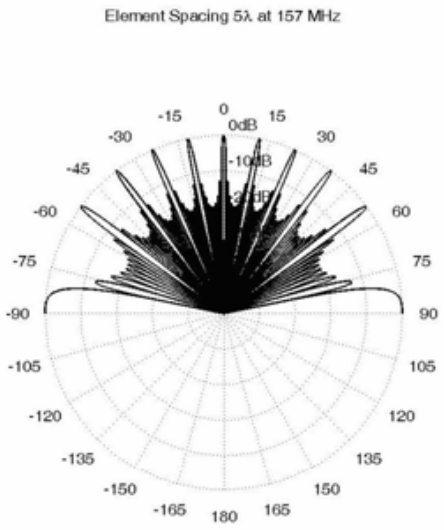


Figure 17. Array pattern of 10 element Uniform Linear Array with 5 times the wavelength spacing of the elements at a frequency of 157 MHz.

A two dimensional planar array is a better representation for the SIGINT/IW sensor network. As seen in Figures 18 – 20, the situation is only marginally better. The simulations indicate that the beamwidth gets narrower, but the number of mainlobes remains consistent. This is consistent with the fact that as the number of elements increase so does the gain and as the gain increases the beamwidth gets narrower assuming that the effective aperture continues to increase as the number of nodes increases.

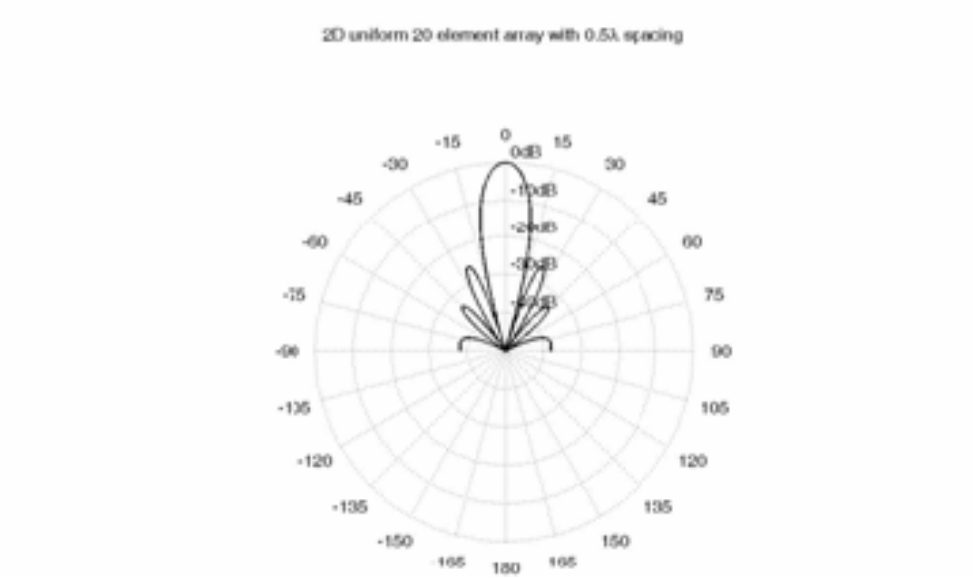


Figure 18. Antenna pattern for a 20 Element 2D Uniform Array with half-wavelength spacing at a frequency of 157 MHz.

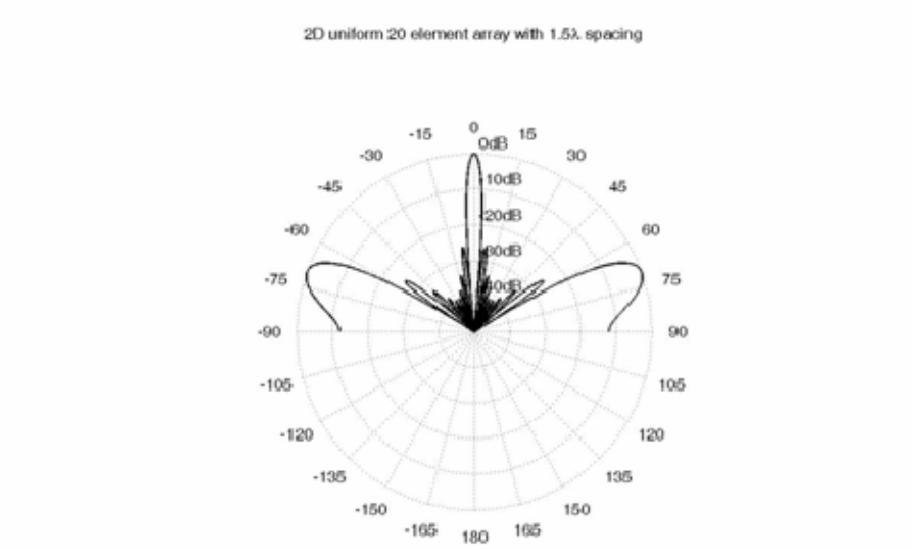


Figure 19. Antenna pattern for a 20 Element 2D Uniform Array with one and half-wavelength spacing at a frequency of 157 MHz.

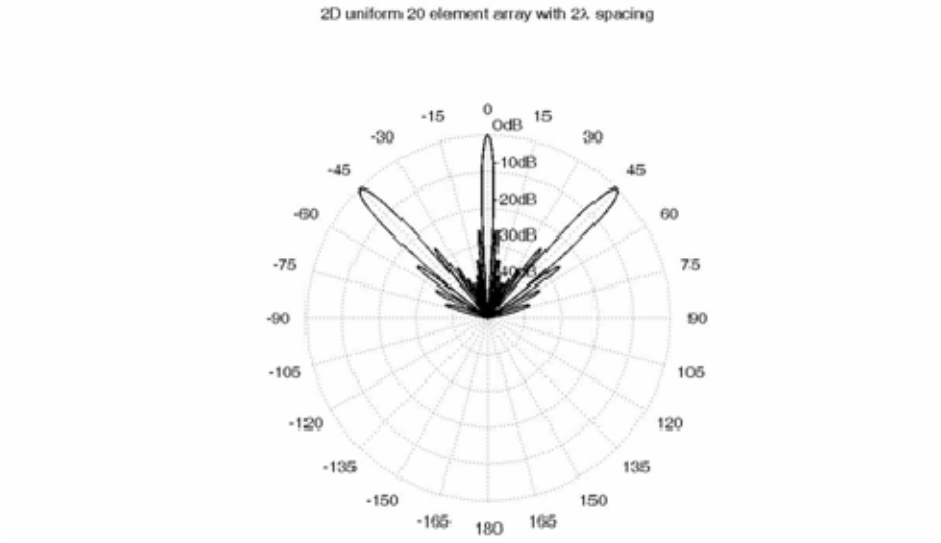


Figure 20. Antenna pattern for a 20 Element 2D Uniform Array two times the wavelength spacing at a frequency of 157 MHz.

As stated before the elements will on average not be spaced that closely together. As seen in Figure 21 a more realistic spacing of about 10 meters clearly shows that the number of mainlobes continues to increase.

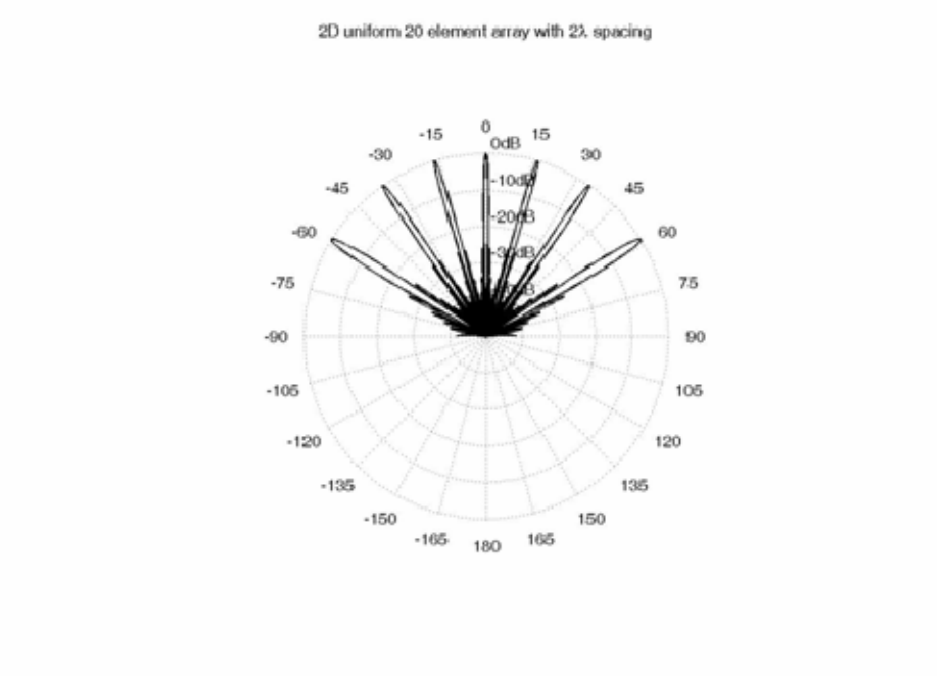


Figure 21. Antenna pattern for a 20 Element 2D Uniform Array with five times the wavelength spacing at a frequency of 157 MHz.

Increasing the number of elements decreases the width of these individual lobes, but does not decrease them in number as can be seen in Figure 22.

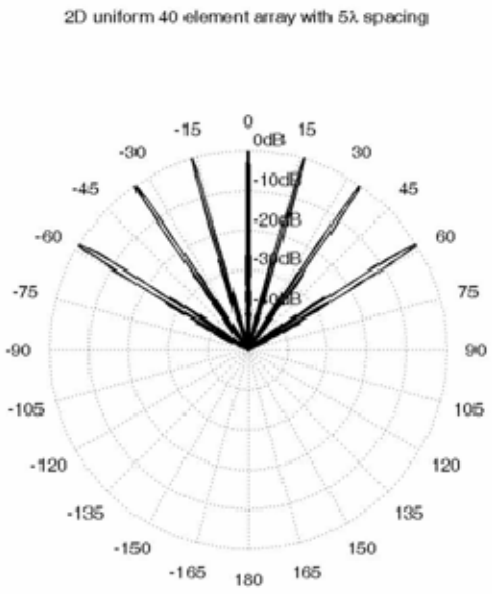


Figure 22. Antenna pattern for a 40 Element 2D Uniform Array with five times the wavelength spacing at a frequency of 157 MHz.

Research suggests that this problem only gets worse as the frequency increases, as can be seen in Figures 23 and 24.

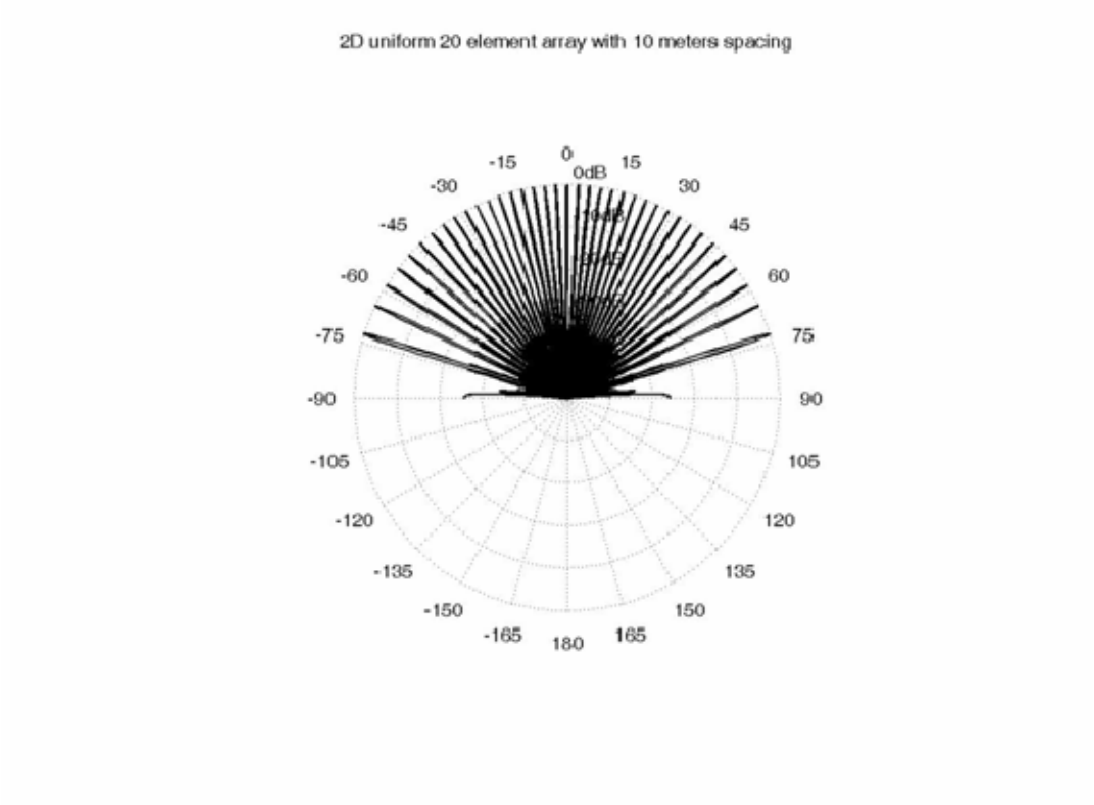


Figure 23. Antenna pattern for a 20 Element 2D Uniform Array with 10 meter element spacing at a frequency of 800 MHz.

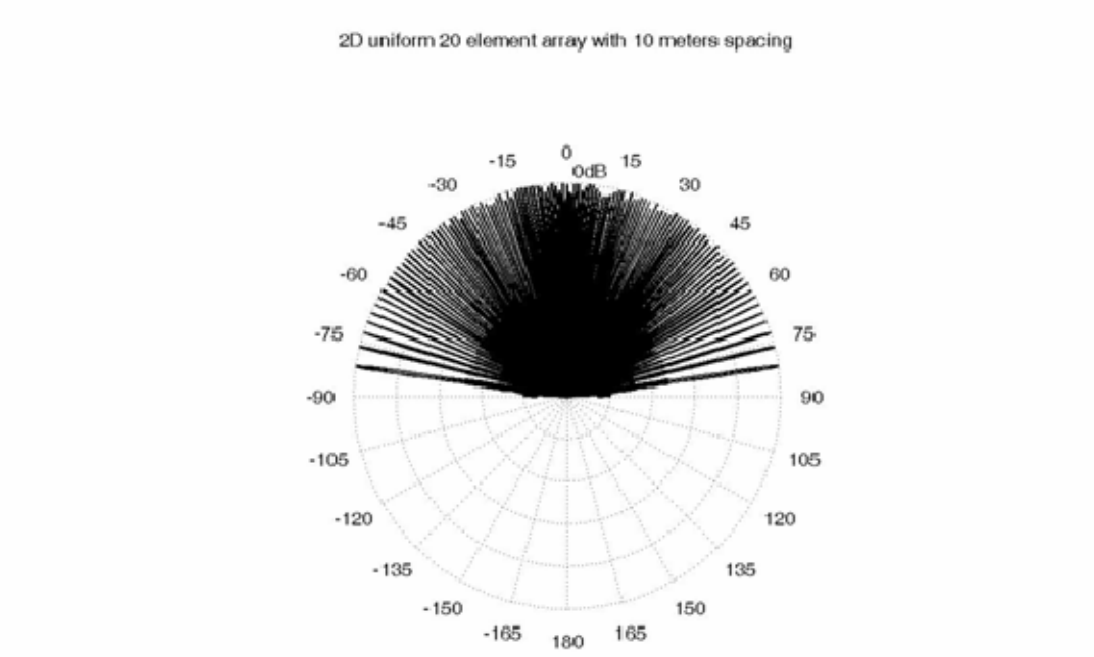


Figure 24. Antenna pattern for a 20 Element 2D Uniform Array with 10 meter element spacing at a frequency of 2.4 GHz.

2. Mutual Coupling

Mutual coupling occurs because the antenna elements within the array interact with each other. This interaction between elements results in an impedance change as seen by each element, which in turn affects the current magnitude, phase and distribution on other neighboring elements [53, 60].

In general, the element spacing should be designed to avoid grating lobes and reduce the adverse effects of mutual coupling. According to [60], the spacing is recommended to be between $\lambda/3$ and $\lambda/2$.

Due to the random distribution of our sensors, the distances between antenna elements will be such that mutual coupling should not be a problem. However, grating lobes will be something that will have to be mitigated as node densities get less. The

sensor grid will in some cases be spaced at distances larger than $\lambda/2$. The choice of spectral estimation and direction-of-arrival technique will need to be capable of coping with this reality.

3. Beam Squint

Phase shifters are designed to adjust phase shifts that may occur during transmission to a desired center frequency f_0 and, if the signal that is being received is not at f_0 as may be the case for wideband signals, an effect called "beam squinting" will occur. An example taken from [40], of beam squinting is shown in Figure 25, where the beam is first pointed to 20 degrees and the frequency is then changed to 1.7 GHz, .8 GHz and 2.5 GHz. It can be seen that the scan angle of the main beam decreases for frequencies higher than the center frequency and increases for frequencies lower than the center frequency. The beamwidth also becomes narrower at higher frequencies and wider for lower frequencies. It can be seen that when the operating frequency changes, then the main beam will steer off from the desired direction, changing both in width and direction with frequency [49]. As stated in [40, 49], the beam squint is greater for a larger array aperture than a smaller array aperture.

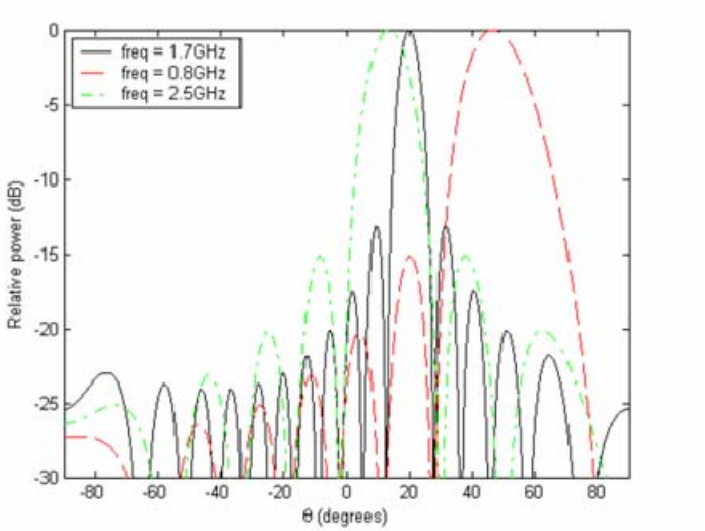


Figure 25. Beam patterns for a phased array at 0.8 GHz, 1.7 GHz and 2.5 GHz when phase shifters are set to steer beam to 20 degrees at 1.7 GHz. [From Ref. 40]

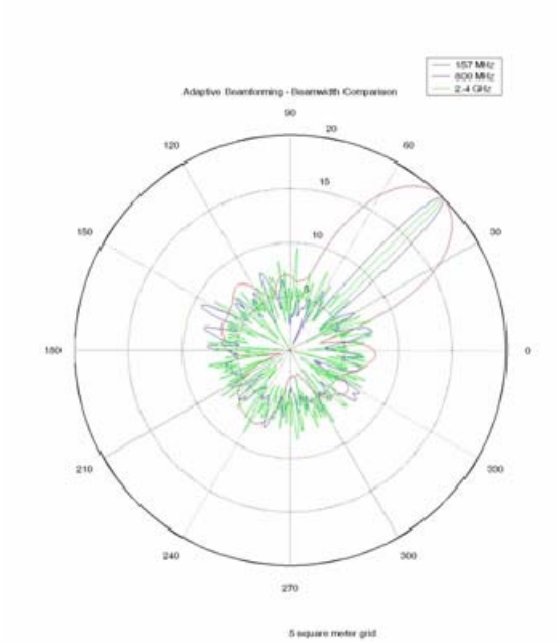


Figure 26. Array pattern for a phased array at 157 MHz, 800 MHz and 2.4 GHz using adaptive beamforming.

In Figure 26, where the beam is first pointed to 45° and the frequency is then changed to 157 MHz, 800 MHz and 2.4 GHz. It can be noted that by using an adaptive beamforming algorithm that the scan angle does not drift, however the beamwidths are still different. The beamwidth also becomes narrower at higher frequencies and wider for lower frequencies.

The above examples were for the narrowband case, but references [40, 49] suggest that it is possible for a phased array to achieve wideband performance by changing the settings of the phase shifters whenever the frequency of a signal with narrow instantaneous bandwidth is changed. This is equivalent to radiating multiple narrowband signals one at a time over a wide range of frequencies by adjusting the settings of the phase shifters.

B. ARRAY PROPERTIES

1. Linear Array

Now a more in depth discussion of array properties will be presented. Continuing with the narrowband example, the array depicted in Figure 13 has been extended to a $Mx1$ array shown in Figure 27. The $Mx1$ array shown in Figure 27 is located in the far field of a point source. The point source (target signal) arrives at an elevation angle with respect to the array normal (i.e., z-axis). For convenience, the reference (central controller) is taken at the origin, and it is assumed that the wavefront arrives at the element t_m seconds before it reaches the origin. Hence, the signal arriving at the element leads the signal arriving at the origin by [73]

$$t_m\left(\theta_a\right) = \frac{x_m \cos\theta_a}{c}, \tag{9}$$

where c is the speed of light, θ_a is the direction of arrival and x_m is the x-coordinate of the m^{th} element.

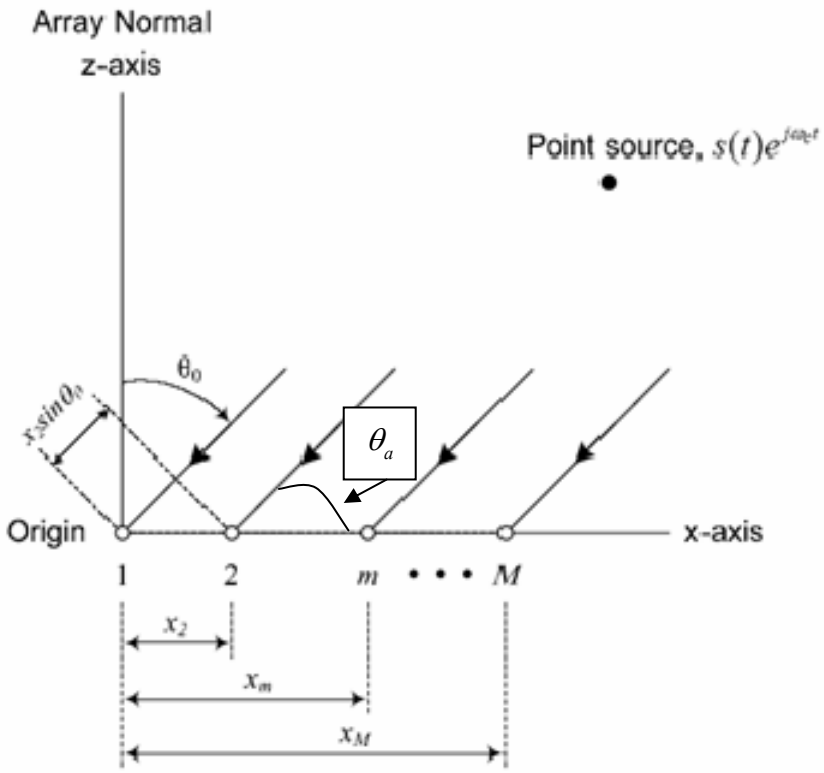


Figure 27. An M×1 omni-directional antenna array. [After Refs. 60,71]

For wideband signals, the parameters in Equation (9) vary with frequency, so now the array factor is given by

$$\Psi(\theta,\omega) = \sum_{m=1}^{M} I_m e^{-j\omega t_m(\theta_a)} e^{j\omega t_m(\theta)}, \tag{10}$$

where $\omega = 2\pi f$ is the center frequency of the signal and I_m is the element weight amplitude [49].

If the frequency of operation changes, then a new set of phase shift settings is required to keep the beam location fixed. This means the first exponent in Equation (10) must change with frequency in the same way that the second exponent changes with frequency.

It can be seen from Equation (9) that $t_m(\theta_a)$ is determined by the angle of arrival of the desired signal indicated by θ_a and the element's position indicated by x_m. For any arbitrary angle θ, the signal arriving at the m^{th} element leads the signal arriving at the origin by

$$t_m(\theta) = \frac{x_m \cos\theta}{c}, \tag{11}$$

where, continuing with the narrowband example, the ω dependence has been dropped. The mutual coupling effects have been ignored. The array factor of the is obtained by adding all the array element outputs together giving [60,73]

$$\Psi(\theta) = \sum_{m=1}^{M} I_m e^{-j\omega_c t_m(\theta_a)} e^{j\omega_c t_m(\theta)}, \tag{12}$$

where I_m and $e^{-j\omega_c t_m(\theta_a)}$ are the magnitude and phase of the current induced on the m^{th} element. Together, I_m and $e^{-j\omega_c t_m(\theta_a)}$ form what is commonly known as the complex weights. Notice that the phase reversal in the term $e^{-j\omega_c t_m(\theta_a)}$ is required in order to create a maximum value of $\Psi(\theta)$ at $\theta = \theta_a$. Hence, the maximum value of of the magnitude $\Psi(\theta)$ occurs at $\theta = \theta_a$ and the main lobe points towards θ_a. Equation (8) can be written in the form given by [60, 73]

$$\Psi(\theta) = \sum_{m=1}^{M} I_m e^{-j\beta(x_m \sin\theta_a)} e^{j\beta(x_m \sin\theta)}, \tag{13}$$

where $w_m = I_m e^{-j\beta(x_m \sin\theta_a)}$ is the complex weight applied to the m^{th} element. The far-field power gain is given by

$$\left|F(\theta)\right|^2 = \left|\sum_{m=1}^{M} I_m e^{-j\beta(x_m \sin\theta_a)} e^{j\beta(x_m \sin\theta_a)}\right|^2, \tag{14}$$

where $F(\theta)$ is the array pattern with isotropic elements defined earlier. If the $Mx1$ array has the main lobe is pointing at $\theta = \theta_a$, the maximum array factor is computed to be

$$\Psi(\theta_a) = \sum_{m=1}^{M} w^*_m e^{j\beta(x_m \sin\theta)} = \sum_{m=1}^{M} I_m e^{-j\beta(x_m \sin\theta_a)} e^{j\beta(x_m \sin\theta_a)} = \sum_{m=1}^{M} I_m = M \,, \qquad (15)$$

using Equation (13) while the power gain is calculated using Equation (14). Therefore, the power gain of a uniform array can be increased by increasing the number of elements. This implies that more sensor nodes can be used within a sensor cluster to produce an antenna beam with higher power gain and hence increase the detection range and offensive information operations capability between the sensor network and the target.

To illustrate the main characteristics of the antenna beam, a plot taken directly from [73] is utilized. The plot of the normalized power gain G_n (in dB) versus θ (in degrees) is shown in Figure 28 and was generated by a 10×1 uniformly excited array. The array consists of isotropic elements with fixed element-element spacing $d = \lambda/2$, and the main lobe, as shown in Figure 28 (a) points in a direction perpendicular to the array axis (i.e., x-axis) and can be categorized as a broadside radiation pattern. As shown in Fig. 54 (b), the 3-dB (or half power) beamwidth θ_{3dB} is about 10.2°, and the highest sidelobe level $S_{L\max}$ is about 13.2 dB below the main lobe level.

The 3-dB beamwidth of a uniformly excited broadside array is given by [60, 73]

$$\theta_{3dB} \approx 0.866 \frac{\lambda}{Md} \,, \qquad (16)$$

where a long array (i.e., $Md \gg \lambda$) has been assumed. The 3-dB beamwidth was calculated to be 9.92° using Equation (16), showing reasonable agreement with the 10.2° obtained from the plot. It can be seen from Equation (16) that θ_{3dB} can be controlled by adjusting M. For example, when M is increased by a factor of ζ, θ_{3dB} is reduced by the same factor, resulting in a narrower beam and vice versa. This allows θ_{3dB} to be adjusted by changing the number of sensor nodes required for beamforming during the detection and tracking of UAV. For example, a large θ_{3dB} is usually required to create a large detection region while a small θ_{3dB} is preferred to localize & track the target and perform sustained collection. What is assumed is that when the number of elements increase so does the effective aperture. It will be shown in Chapter VI, Section B,

Subsection 3 that when the physical area is held constant and the number of nodes (elements) is increased the beamwidth actually gets broader and not narrower implying less gain. This is further discussed in sub-section 4.

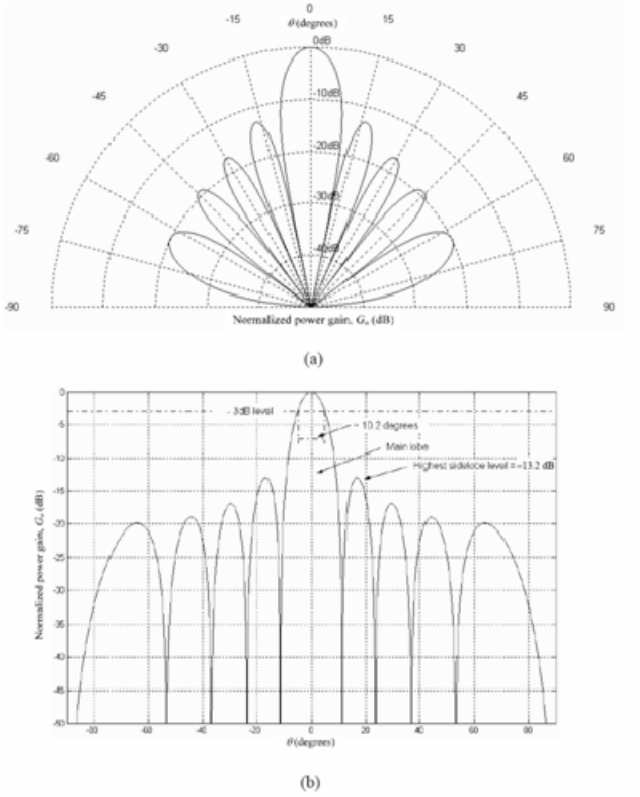


Figure 28. Normalized power gain of the beam generated by a 10×1 uniformly excited square array with isotropic elements, fixed element spacing of $d = \lambda/2$, and $\theta_a = 0^\circ$ (a) Polar plot and (b) Rectangular plot in the X-Z plane indicating the 3-dB beamwidth of 10.2° and the highest sidelobe level of -13.2 dB. [From Ref. 73]

The array factor and the power gain of a one-dimensional array can be extended to a two-dimensional array as presented in the next section.

2. Planar Array

Figure 29 shows an MxN omni-directional antenna array located in the far field of a point source. The received signal arrives at an elevation angle, θ with respect to the z-axis and an azimuth angle, ϕ of with respect to the x-axis. The central controller is taken at the origin, and it is assumed that the received signal arrives at the element t_{mn}

seconds before it reaches the central controller. Therefore, the signal arriving at the $(m,n)^{th}$ element leads the signal arriving at the origin by [60, 73]

$$t_{mn}(\theta_a,\phi_0)=\frac{x_{mn}\sin\theta_a\cos\phi_0+y_{mn}\sin\theta_a\sin\phi_0}{c}, \tag{17}$$

where c is the speed of light and x_{mn} is the x-coordinate and y_{mn} y-coordinate of the $(m,n)^{th}$ element.

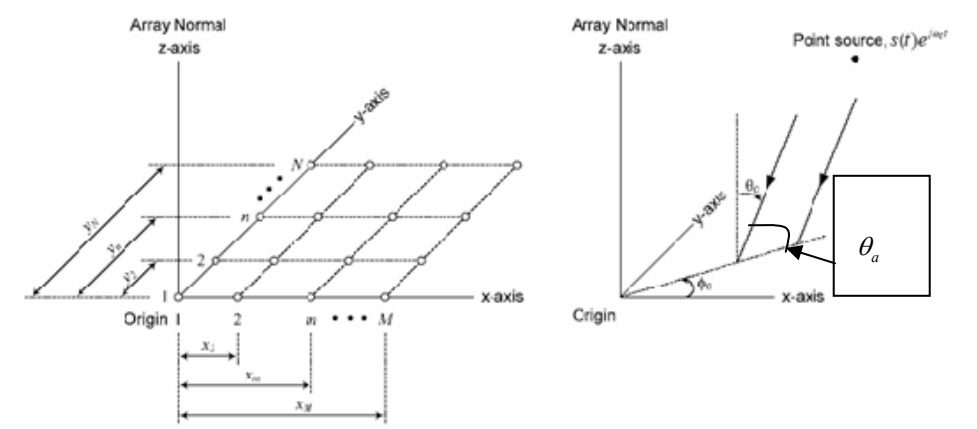


Figure 29. An $M\times N$ omni-directional antenna array. [After Ref. 73]

For any set of arbitrary angles, (θ,ϕ), the signal arriving at the $(m,n)^{th}$ element leads the signal arriving at the origin by

$$t_{mn}(\theta,\phi)=\frac{x_{mn}\sin\theta\cos\phi+y_{mn}\sin\theta\sin\phi}{c}, \tag{18}$$

The two-dimensional array factor is given by

$$\Psi(\theta,\phi)=\sum_{m=1}^{M}\sum_{n=1}^{N}I_{mn}e^{-j\beta\left(x_{mn}\sin\theta_a\cos\phi_0+y_{mn}\sin\theta_a\sin\phi_0\right)}e^{j\beta\left(x_{mn}\sin\theta\cos\phi+y_{mn}\sin\theta\sin\phi\right)}, \tag{19}$$

without considering the effects of mutual coupling among the array elements. [60] Equation (19) can be rewritten as

$$\Psi(\theta,\phi)=\sum_{m=1}^{M}\sum_{n=1}^{N}w^{*}_{mn}e^{j\beta\left(x_{mn}\sin\theta\cos\phi+y_{mn}\sin\theta\sin\phi\right)}, \tag{20}$$

where β is the wavenumber given by $2\pi/\lambda$ and $w_{mn} = I_{mn} e^{-j\beta(x_{mn}\sin\theta_a\cos\phi_0 + y_{mn}\sin\theta_a\sin\phi_0)}$ is the complex weight applied to the $(m,n)^{th}$ element. The maximum value of $F(\theta,\phi)$ occurs at $(\theta,\phi) = (\theta_a,\phi_0)$, and the main lobe points towards (θ_a,ϕ_0), The radiation pattern with isotropic array elements is defined as

$$\left|F(\theta,\phi)\right|^2 = \left|\Psi(\theta,\phi)\right|^2 = \left|\sum_{m=1}^{M}\sum_{n=1}^{N} w^*_{mn} e^{j\beta(x_{mn}\sin\theta\cos\phi + y_{mn}\sin\theta\sin\phi)}\right|^2, \tag{21}$$

Consider an MxN uniformly excited array with isotropic elements and the main lobe pointing at $\theta_a = \phi_0 = 0^\circ$ The maximum array factor and the power gain are (NM) and $(NM)^2$, respectively. This result is similar to the linear array. The same conclusions can be drawn that the power gain of a planar array can be increased by increasing the number of elements with the assumption that the effective aperture increases.

The following example taken from [68] illustrates the affects of different array sizes on the beamwidth of the beam. The beam pattern formed by 3×3, 5×5 and 7×7 arrays are shown in Figure 56 in which plots were generated for $\phi = 45^\circ$. These uniformly excited square arrays consist of isotropic elements with a fixed element-element spacing $d = \lambda/2$. The main lobe for all the three arrays is assumed to point at $(\theta_a,\phi_0) = (0^\circ, 45^\circ)$ as illustrated in Figure 30(a). Figure 30(b) shows that θ_{3dB} increases from 15° to 37° when the number of array elements decreases from 7×7 to 3×3. Therefore, the antenna beam of a uniformly excited array with isotropic elements and fixed element-element spacing becomes broader when the number of elements decreases. This observation is consistent with the conclusion in the previous section for the one-dimensional case. Section 4 will investigate the beamwidth affects from a randomly distributed network.

In summary, the array factor and power gain equations for the planar array were presented in this section, and it was concluded that they were affected by the azimuth of the target signal, the node density, the node spacing, and the complex weights w_{mn}. The main characteristics of the antenna beam, such as 3-dB beamwidth and highest sidelobe

level, were introduced. The effects of the number of array elements on the antenna power gain and the 3-dB beamwidth were also discussed.

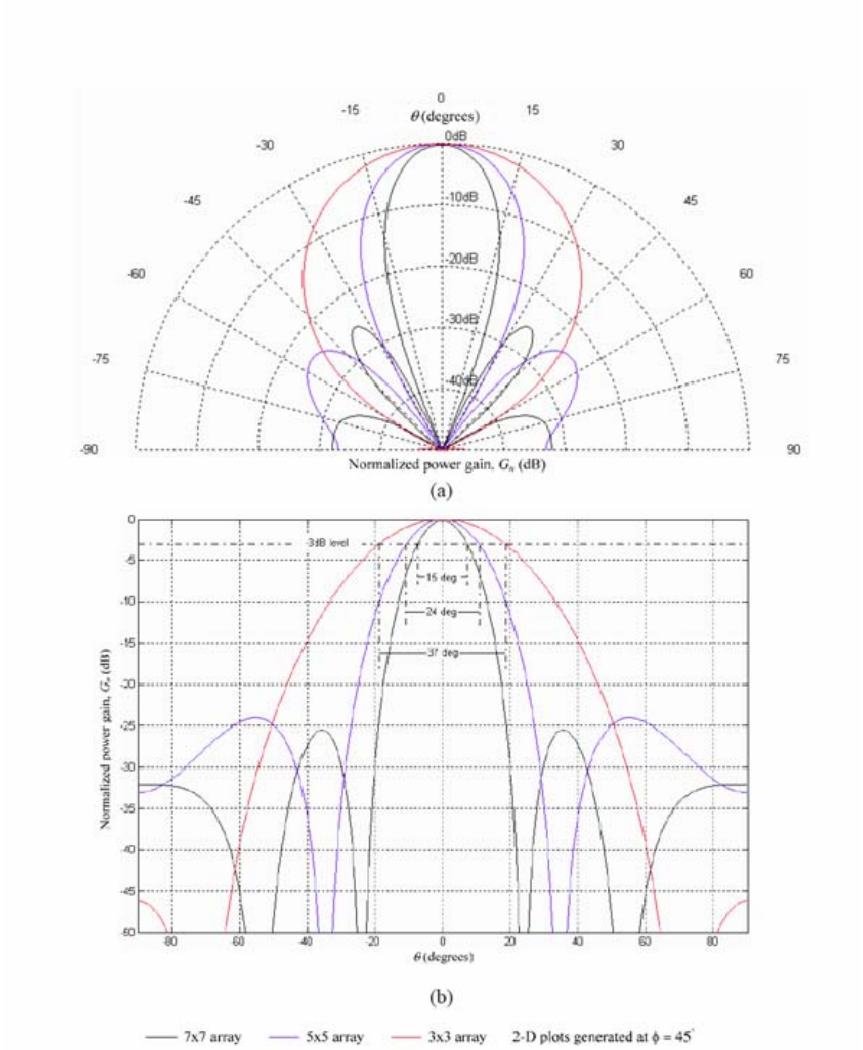


Figure 30. Normalized power gain of the beams generated by 7×7, 5×5 and 3×3 uniformly excited square arrays with isotropic elements, fixed element spacing of $d = \lambda/2$, and $(\theta_a, \phi_0) = (0^\circ, 45^\circ)$: (a) Polar plots and (b) X-Y plots showing the decrease in 3-dB beam-width as the number of elements increase.[From Ref. 73]

3. Random Array

The SIGINT/IW network will consist of SIGINT/IW sensor nodes that are deployed randomly over an area of interest, resulting in random positioning of the sensor nodes. Subsequently, these sensor nodes form clusters and their locations are determined using location discovery techniques and are reported back to the central controller. The random position of the sensor nodes within the sensor cluster gives rise to what is

referred to as a random array. In addition, the SIGINT/IW antenna elements will in some cases not be located within a half-wavelength from each other. Therefore an algorithm capable of mitigating the grating lobes discussed in section A, subsection 1 will have to be employed.

The SIGINT/IW sensor network will be a random array antenna that will adaptively control the pattern in order to enhance collection and/or facilitate offensive information operations in the direction of the target. Once the target has been localized the central controller can enhance the gain toward the desired signals and can null toward undesired signals. Correct control of beams and nulls is indispensable for the adaptive antenna. The pattern of the array will be controlled by dynamically varying the phase and amplitude of the received signal of each element. There are a variety of weight control algorithms for adaptive antennas. This research will build on the LMS algorithm research conducted in [73].

Continuing with the narrowband example from Section 1, each element output $s_i(t)$ is multiplied by a complex weight w_i^*, modifying the phase and amplitude relation between the branches, and summed to give $b(t)$.

$$\Psi(\theta) = \begin{bmatrix} w_1^* & w_2^* & w_3^* & \cdots & w_N^* \end{bmatrix} \begin{bmatrix} 1 \\ e^{-j\beta d\cos(\theta)} \\ e^{-j\beta 2d\cos(\theta)} \\ \vdots \\ e^{-j\beta(N-1)d\cos(\theta)} \end{bmatrix}, \qquad b(t) = w^H s(t), \tag{22}$$

Since the array factor for the overall antenna pattern is dependent on the phase and amplitude relationship between the branches and the weight vector modifies the phase and amplitude relationship, the overall array pattern can be continually modified by adjusting the weight vector [14, 60]. The overall pattern for the array is given by

$$F(\omega,\theta,\phi)=\left|w^{H}\Psi(\omega,\theta,\phi)\right|, \tag{23}$$

by changing the phase in each element the network can steer the beam virtually instantaneously.

4. Controlling the Size of the Beamwidth

Understanding how to control the beamwidth of the beams will be critical. In a search mode a wider beam, i.e. larger detection region is preferable. When sustained collection and/or offensive operations is desirable then a more directive beam with higher gain in the target direction will be desirable. The equations for a two-dimensional planar array taken from [3] will be used as a guide for comparison with the randomly configured wireless antenna array.

The beamwidth for a planar array is determined by using Equations (24) and (25). In order to determine the respective beamwidth the array is separated into *x* and *y* linear arrays. These equations for the respective scan angles for both elevation and azimuth are shown in Equation (26) [3].

$$\Theta_{elevation}=\sqrt{\left(\frac{1}{\cos^{2}(\theta_{a})\left[\theta_{x_a}{}^{-2}\cos^{2}(\phi_{0})+\theta_{y_a}{}^{-2}\sin^{2}(\phi_{0})\right]}\right)}, \tag{24}$$

$$\Theta_{azimuth}=\sqrt{\left(\frac{1}{\left[\theta_{x_a}{}^{-2}\sin^{2}(\phi_{0})+\theta_{y_a}{}^{-2}\cos^{2}(\phi_{0})\right]}\right)}, \tag{25}$$

where ϕ_0 is the desired azimuth scan angle and θ_a is the desired elevation angle. It can be seen how the beamwidth in the azimuth and elevation directions impacts the gain of the beam. [3]

$$\begin{aligned}\theta_{x_a}&=\cos^{-1}\left[\cos\theta_{a}-0.443\left(\frac{\lambda}{(Length+d)}\right)\right],\\ \theta_{y_a}&=\cos^{-1}\left[\cos\theta_{a}+0.443\left(\frac{\lambda}{(Length+d)}\right)\right]\end{aligned}, \tag{26}$$

where $d = Length / N$, *Length* is the length of the uniform linear array and *N* is the number of nodes along the length of the uniform linear array. Now shown below for the azimuth [3].

$$\begin{aligned} \phi_{x_0} &= \cos^{-1}\left[\cos\phi_0 - 0.443\left(\frac{\lambda}{(L+d)}\right)\right], \\ \phi_{y_0} &= \cos^{-1}\left[\cos\phi_0 + 0.443\left(\frac{\lambda}{(L+d)}\right)\right] \end{aligned}, \tag{27}$$

where $d = Length / N$, *Length* is the length of the uniform linear array and *N* is the number of nodes along the length of the uniform linear array. It will be shown that as the number of elements increases for a specific area, i.e. the node density increases, the beamwidth reaches an upper-bound for a given wavelength [3].

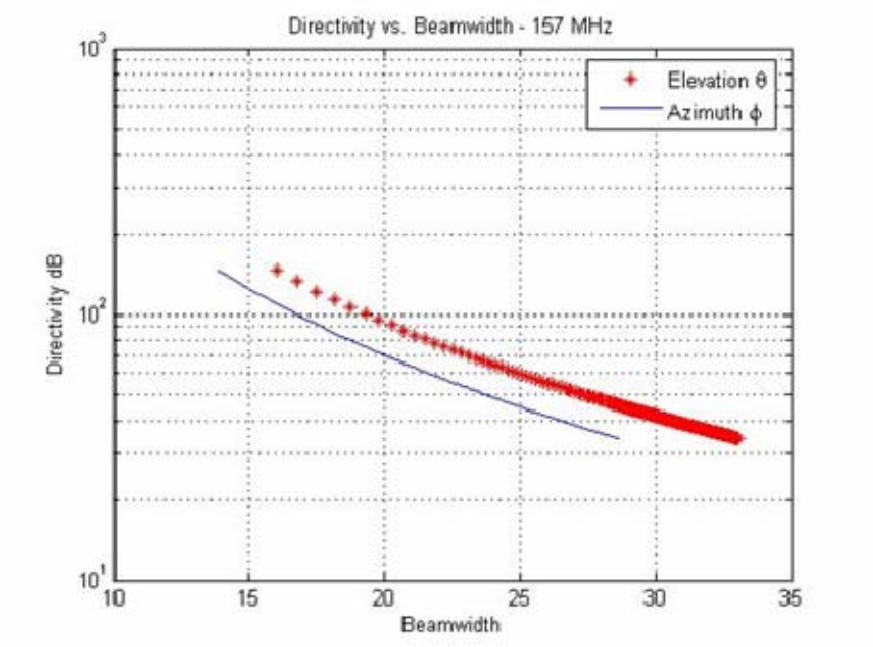
Likewise it can be seen from Equations (26) and (27) that as the length of each *x* and *y* increases for a given number of nodes that the beamwidths get narrower. Equation (28) is for the solid beam of the array using the azimuth and elevation calculations from Equations (24) and (25). Equation (29) is for the directivity, gain in a particular direction, which is based on the solid angle of the array. It can be seen from Equations (28) and (29) that as the beamwidths get narrower the directivity goes up increasing the gain in that particular direction and vice-versa when the beamwidths increase the directivity in a particular direction goes down.

Armed with those two vital facts the central controller can now control the width of the beam by selecting areas of the overall grid that contains a sufficient node density in order to produce the appropriate beam size. This feat can be accomplished simply by choosing nodes that are contained within a specific area.

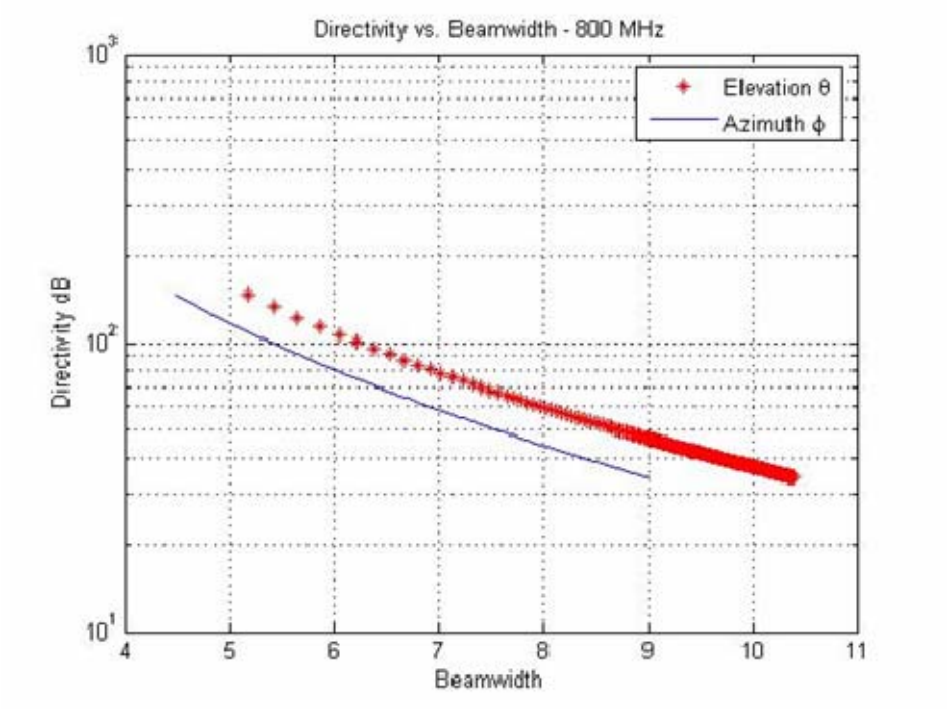
$$\Omega_A = \Theta_{elevation}\Theta_{azimuth}, \tag{28}$$

$$D \approx \frac{32400}{\Omega_A(\text{degrees})}, \tag{29}$$

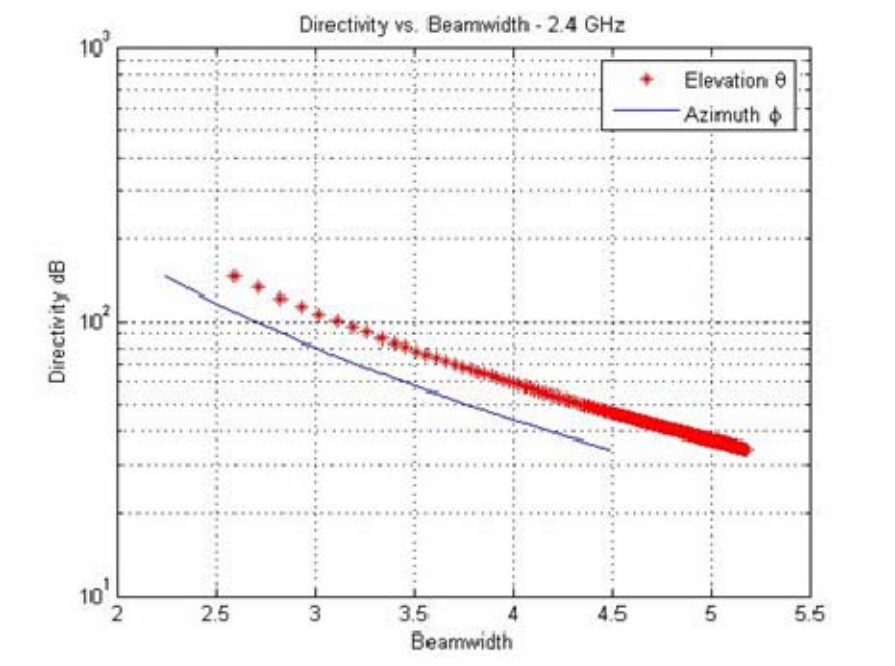
The inverse relationship between directivity and beamwidth can be observed by analyzing Equation (29). Figures 31 (a) – (c) show this inverse relationship between directivity and beamwidth for three frequencies: 157 MHz, 800 MHz, and 2.4 GHz. The three frequencies were distributed over different areas, $A_2 = 25m^2$, $A_2 = 9m^2$, $A_2 = 4m^2$, respectively to account for the differences in wavelength. The simulations ignore any mutual coupling effects that would occur as the baseline, d, became less than $\lambda/3$. This relationship is important because for a search mode a larger detection region, i.e. larger beamwidth, is desirable. The ability of the central controller to control the beamwidth will be an important operational constraint.



(a) 157 MHz



(b) 800 MHz.



(c) 2.4 GHz

Figure 31. The inverse relationship between directivity and beamwidth. 250 nodes were distributed over three different areas depending on the frequency tested, (a) Area, $A_2 = 25$ m^2 for frequency = 157 MHz; (b) $A_2 = 9$ m^2 for frequency = 800 MHz; and (c) $A_2 = 4$ m^2, for frequency = 2.4 GHz.

For a given SIGINT/IW network with K sensors spread over an area, A_1, the average distance between the sensor nodes will increase as the central controller creates an array containing Λ sensors contained with an increasing area, A_2, to begin beamforming. As can be seen in equations (24) – (27) as the average distance increases between sensor nodes this causes a decrease in node density and a resulting decrease in the beamwidth. This relationship is illustrated in Figures 32-34.

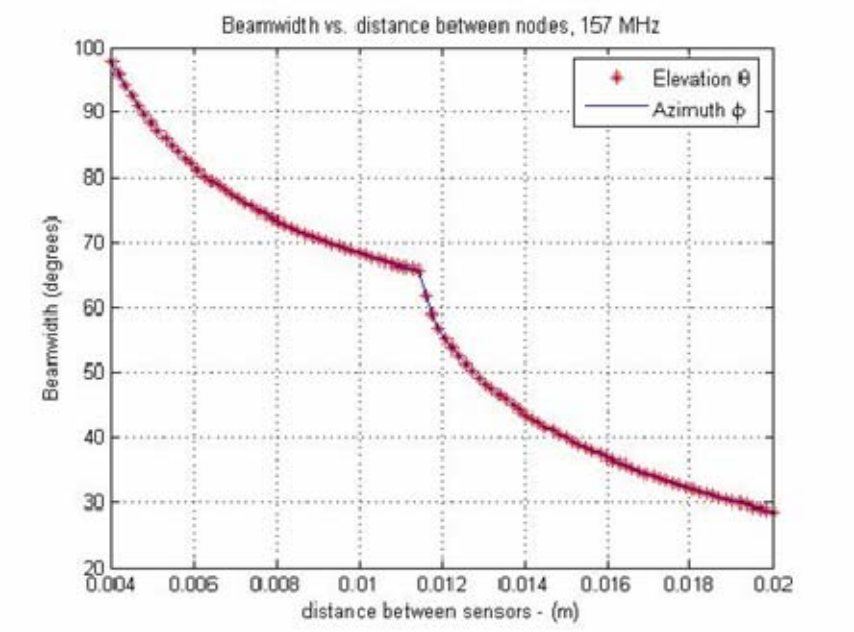


Figure 32. The relationship between beamwidth and node density. The Average distance between nodes increases as node density decreases. Nodes are contained with an Area, A_2 = 25 square meter area.

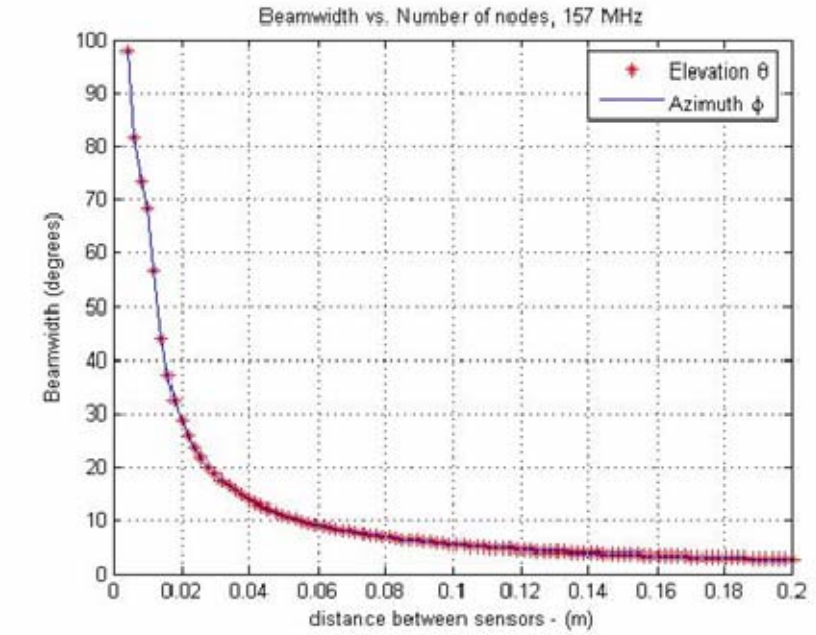


Figure 33. The relationship between beamwidth and node density. The Average distance between nodes increases as node density decreases. Nodes contained within an Area, A_2 = 2500 square meter area.

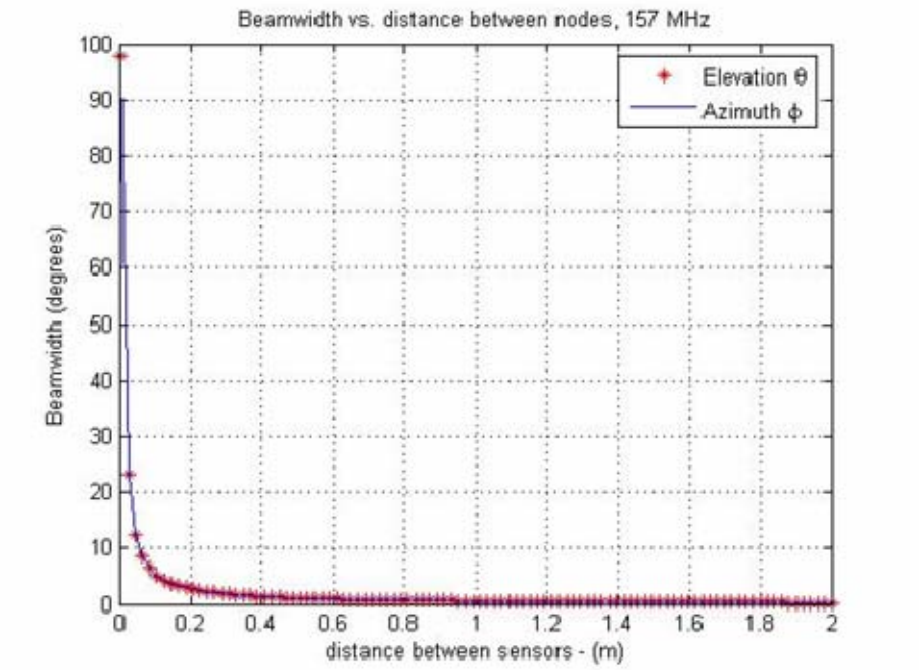


Figure 34. The relationship between beamwidth and node density. The Average distance between nodes increases as node density decreases. Nodes contained within an Area, A_2 = 250000 square meter area.

Simulations in Chapter VI, Section B will be used to illustrate how the beamwidth of the mainbeam is affected by the central controller. The central controller can affect the mainbeam by changing the node density through its selection process. Specifically, as the number of nodes is held constant and the central controller selects from an array of nodes, Λ, that are dispersed within a smaller area, A_2, the beamwidth of the mainbeam can be widened for search. In effect the central controller is simply selecting the nodes that are closer in distance to itself. Likewise, as the central controller selects from an array of nodes, Λ, that are dispersed within a larger area, A_2, the beamwidth of the mainbeam can be narrowed for sustained collection. In effect the central controller is selecting nodes that are located further from its location.

In the above case the effective area remains constant and SIGINT/IW sensors were added in effect increasing the node density. Since the node density increases and the effective area remains unchanged the baseline distance, d, between nodes decreases. The larger the baseline, d, between nodes results in a narrower beamwidth which is similar to the previous sections where the nodes become spread over a greater area, A_2, then the beamwidth gets narrower. Therefore as the node density increases the baseline, d, becomes smaller resulting in a broader beam, which can be seen in Figures 32-34. This is illustrated quite well in Equations (26) – (27) that as the number of nodes is increased, Equations (25) – (26) asymptotically reach a constant value. This will be illustrated in Chapter VI, Section B.

It can be assumed that a reasonable node density will be required in order to do adaptive beamforming. As will be shown in Chapter IV, Section B this node density is highly dependent on the frequency of the signal-of-interest. This highlights the limitations of this type of network in using beamforming to enhance collection at higher frequencies. The node densities required for enhanced collection would be high, which in turn would require the sensors to be small to maintain some level of covertness. This relationship will be further discussed in Chapter IV, Section B.

It will be shown in Chapter VI that the beamwidth of the beam can be controlled by changing the subset selection area of the wireless SIGINT/IW antenna network. For our sensor grid, the central controller selects the number of array elements needed to

form the desired beam. The central controller in effect creates an array, which is merely a subset of the total array. For example, consider a sensor grid that contains K elements randomly spread over a total area, A_1. The central controller will select nodes from an area A_2 containing Λ elements randomly spread throughout the area, where $\Lambda \leq K$ and $A_2 \leq A_1$. Of the nodes within area, A_2, P nodes will be chosen, where $P \leq \Lambda$. The P node subset will form the basis for the array factor. The area A_2, in effect becomes the effective aperture of the antenna grid. Based on the size of the area, A_2, the beamwidth of the beam can be tailored to specific applications, specifically, a wide beam for detection and a narrow beam for localization and/or collection.

C. SPECTRAL ESTIMATION

The periodogram and the Multiple SIgnal Classification (MUSIC) algorithms were investigated for spectral estimation. After the sensor nodes are deployed, they are grouped into clusters with a primary controlling node, the central controller, being established. Once the location of the sensor nodes has been determined by the central controller spectral estimation will be initiated and coordinated in order to evaluate the spectrum and determine the frequency of the SOI. An introduction into the two spectral estimation techniques investigated will be presented below [16, 73].

Spectral estimation methods estimate the frequency of the SOI impinging on our distributed antenna array. One of the more straightforward methods is that of the Periodogram. This method is widely used because the computation is efficient, as shown in Equation (30) [66]. In practice, a fast Fourier transform (FFT) is used to compute the spectrum of the given data sequence.

$$\hat{P}_S\left(e^{j\omega}\right) = \frac{1}{N_s}\left|S\left(e^{j\omega}\right)\right|^2, \tag{30}$$

where $S\left(e^{j\omega}\right)$ is the Fourier transform of the received signal and N_S is the number of samples used in the Fourier transform.

The MUSIC method was developed by Schmidt, *et al*.[47] This algorithm will be presented in more detail in Chapter V, Section A, because this method will also be investigated for determining angle-of-arrival of a signal of interest (SOI).

The central controller will be assumed to have more processing power and a higher gain antenna than the individual nodes. Therefore, the central controller will be considered the reference antenna. The research was conducted utilizing built-in MATLAB® functions to compare the two methods.

Both of these techniques have advantages and disadvantages. Research indicates that both perform adequately at lower frequencies, but MUSIC continues to give sharper results at high frequencies as seen in Figures 35 – 37.

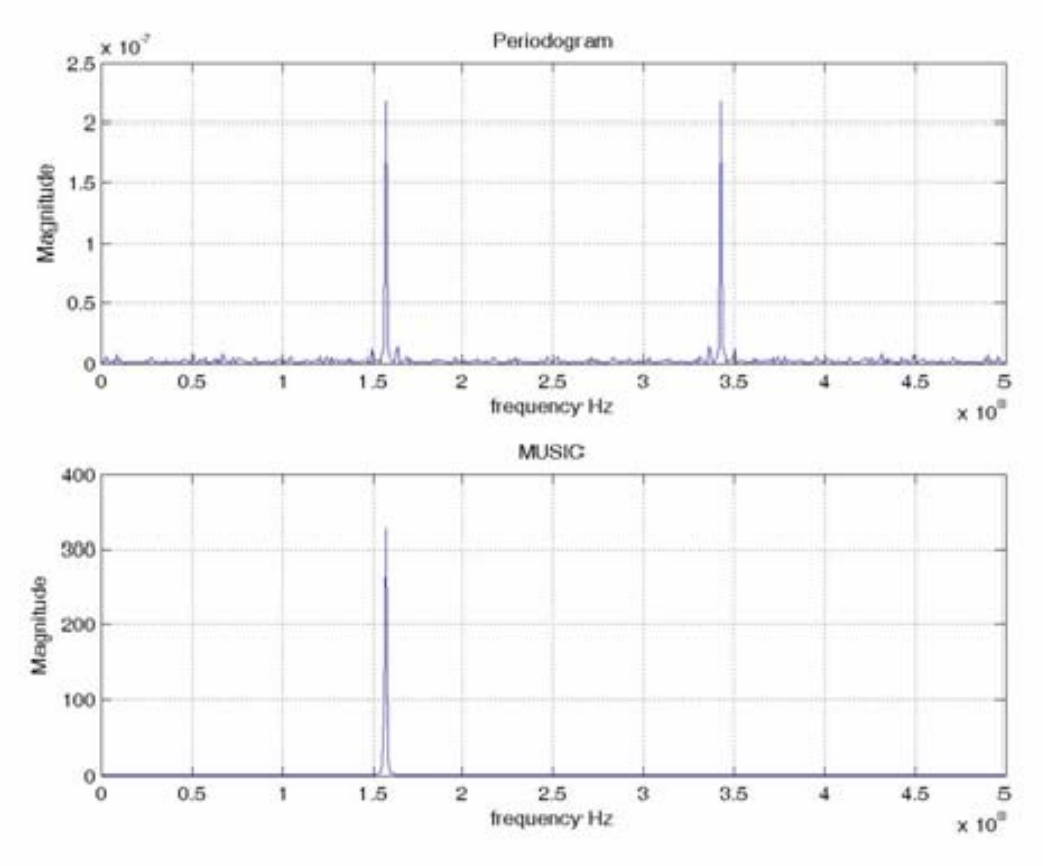


Figure 35. Spectral Estimation results 157 MHz.

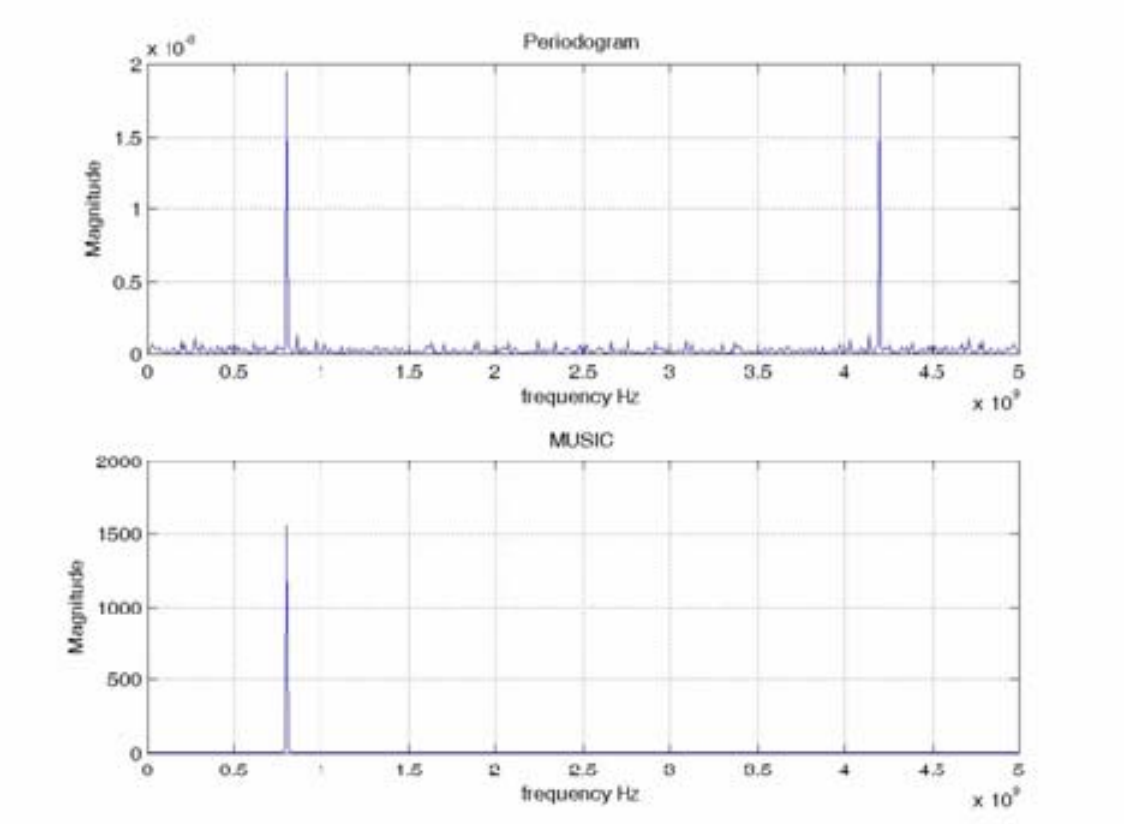


Figure 36. Spectral Estimation results 800 MHz.

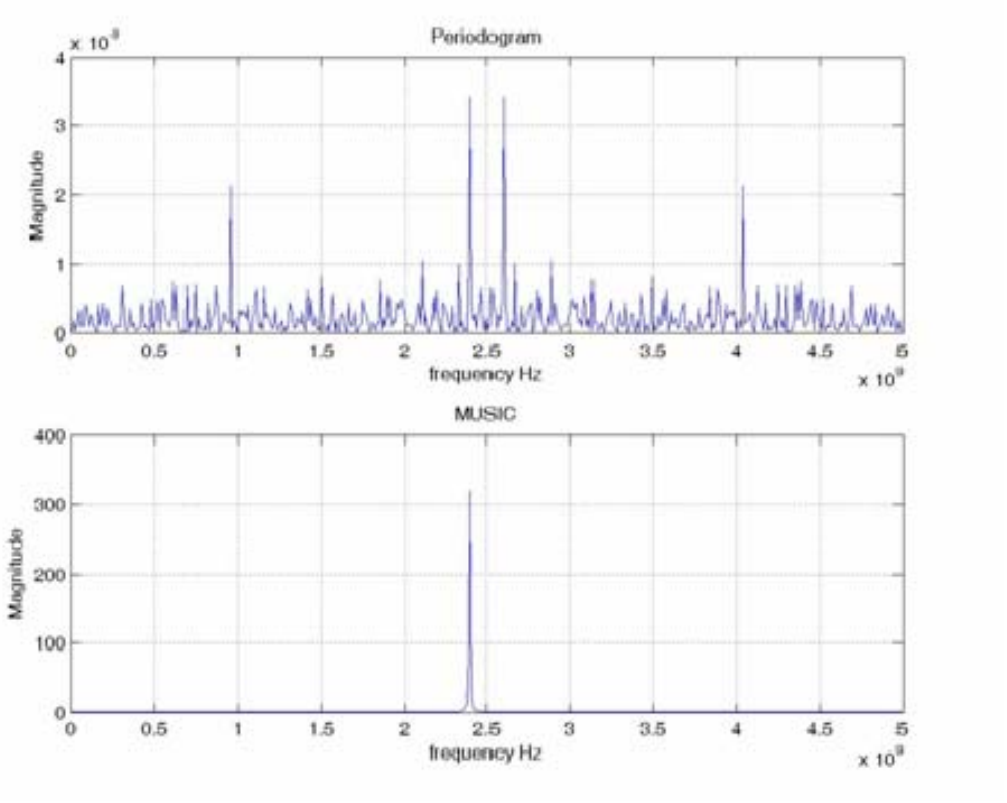


Figure 37. Spectral Estimation results 2.4 GHz.

As discussed earlier the periodogram takes the Fourier transform of the received power spectral density. As can be seen in the figures, the noise appears to have more of an effect at the higher frequencies. Where as in the case of MUSIC the signals are chosen by selecting eigenvalues that are orthogonal to the noise subspace, the effect of the noise has been removed. The estimates of the frequencies are taken as those values of ω that create peaks in the spectrum. When the ensemble average of the array input covariance matrix is known and the noise can be considered uncorrelated and identically distributed between the elements, the peaks of the MUSIC spectrum are guaranteed to correspond to the frequencies of the signals incident on the array [16, 30, 47, 58, 62, 66, 69, 71, 73].

MUSIC gives better resolution for signals operating at closely spaced frequencies. MUSIC is highly robust, but it requires characterization of the array response. The periodogram has the advantage of being extremely easy to implement in FPGA hardware allowing for a very fast solution. However, the periodogram is unable to resolve signals spaced less than $2\pi / N$ apart [62, 66]. As described in [62], the resolution limit for the

MUSIC method is more complex, and depends on the number of samples as well as SNR and number of antenna elements. For more detailed descriptions on the assumptions on the resolution limits of MUSIC see [62].

However, MUSIC does not perform well in practical applications where the number of signals present are unknown. The algorithm is unable to separate the signal and noise into their respective subspaces. Due to its simplicity and wide use it is assumed that the SIGINT/IW network will employ some variant of the periodogram to determine the frequency of the SOI [62].

D. ELEMENT SYNCHRONIZATION FOR THE SIGINT/IW SENSOR GRID

An investigation was conducted into the two synchronization methods developed at NPS. The intent of this investigation is to explore the viability of the methods to synchronize the elements contained within the SIGINT/IW network. In [28], the "brute force" and "beam tagging" techniques were applied to a wireless radar scenario. The wireless Tx/Rx modules were distributed randomly over the hull of a DDx Destroyer. The MATLAB® code created for these simulations was altered to investigate the potential opportunities for use within a SIGINT/IW System. This modified code can be found in Appendices A and B. The random distribution of sensors in [28] was bounded by the dimensions of the DDx. The code was enhanced to allow for the random distribution of SIGINT/IW sensors over a generic random surface. The simulations in MATLAB® were performed to verify the "brute force" and "beamtagging" techniques. The programs were used to phase synchronize with 400 elements, distributed randomly. The central controller was located at the origin so that the transmission path length was equal to the norm of the x and y coordinates of element n.

1. "Brute Force" Synchronization Technique

As was previously shown in Figure 9, the brute force technique functions as follows. All elements are initialized with zero phase and the element closest to the origin is selected as the reference element. Each element is selected in turn for synchronization. When the element is selected, its phase shifter is incremented in 22.5° steps, which is equivalent to four-bit digitization, until the combined field is minimum. This is then

repeated for the rest of the elements. Four-bit digitization was deemed satisfactory for digital phase shifter quantization based on the required sidelobe level [28].

Figure 38 shows the phase error of each element from the reference element plotted against the number of iterations. A change in color denotes a new element being synchronized. For one realization of the 400 randomly located elements, 3247 iterations were required to perform synchronization. The frequency chosen was 400 MHz, which was the same used in [28] in order to get a good baseline comparison.

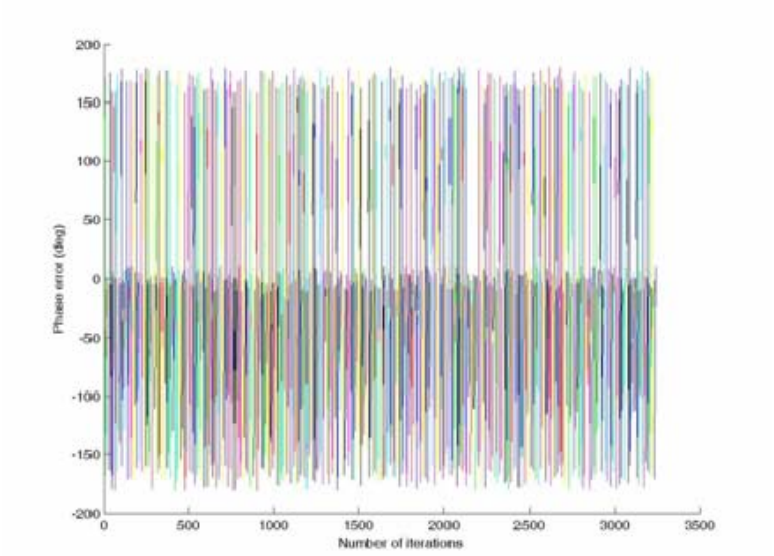


Figure 38. Phase Synchronization of 400 Elements using the Bruteforce Technique.

For fixed phase steps, it is not possible to achieve complete cancellation. Therefore the brute force technique requires a threshold value to detect the minima when the two signals cancel. Assuming equal amplitudes and using 22.5° steps, the final phase error is $\pm 11.5^{\circ}$. Using phasor geometry the minimum can be determined. A threshold of 0.2 (-14 dB) was used in [28] to determine the minima. With *16* phase steps between 0° and 360° about eight iterations on average were required to synchronize each element. When the elements are synchronized with the reference element, the final phase error is between $\pm 11.25^{\circ}$ as shown in Figure 39. These are the same results obtained in [28] showing that the algorithm is adaptable to a totally randomly distributed antenna grid.

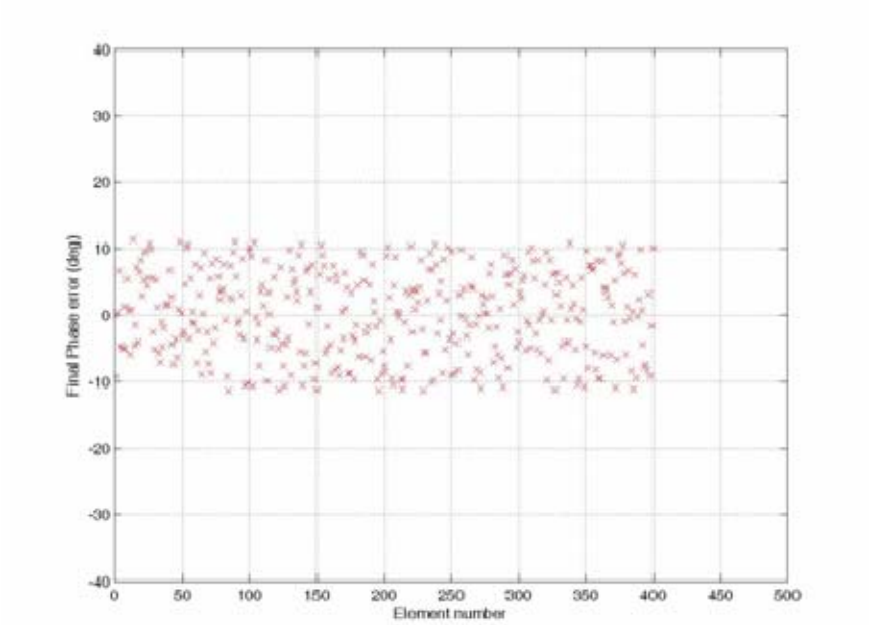


Figure 39. Final Phase Error of all Elements after Phase Synchronization using the "brute-force technique.

Now the frequency was changed to the three frequencies utilized throughout this research, 157 MHz, 800 MHz, and 2.4 GHz, to quantify the effects. As seen in Figures 68 – 85 the frequency being synchronized did not seem to have an effect. However, these simulations do not account for channel characteristics, i.e., multipath, etc. that could be a factor, especially at high frequencies. The phase error for all cases was the same $\pm 11.25^{\circ}$. The figures highlight the synchronization and final phase errors of individual nodes instead of the entire array.

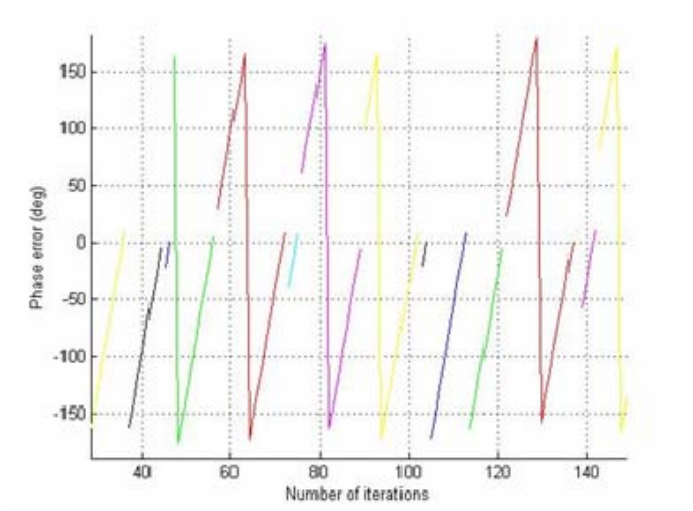


Figure 40. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 157 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 25 square meter area.

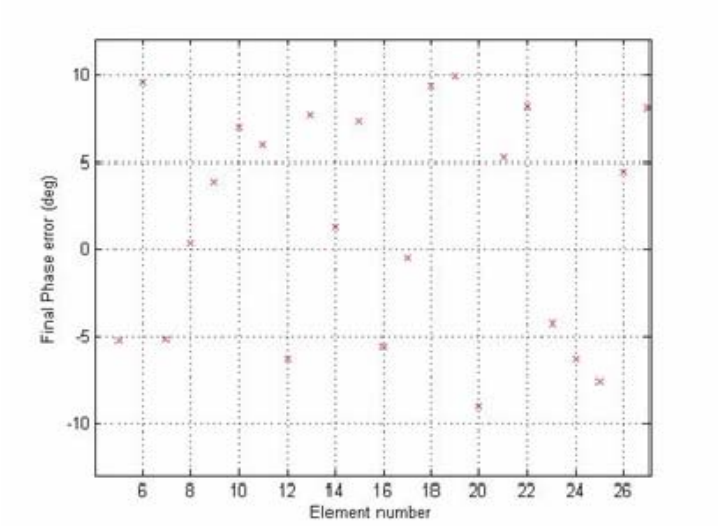


Figure 41. Final phase error for a sample of nodes from a collection of 400 nodes in a 25 square meter area. Frequency of the SOI is 157 MHz.

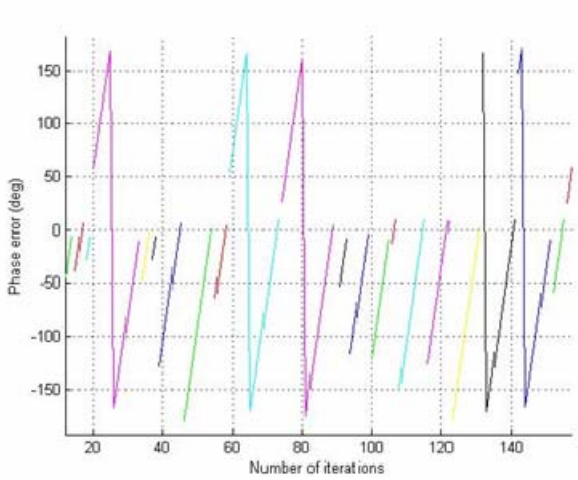


Figure 42. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 157 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 2500 square meter area.

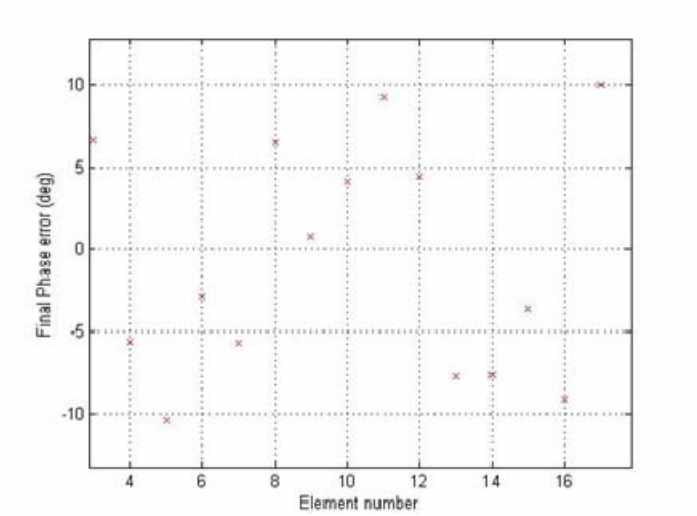


Figure 43. Final phase error for a sample of nodes from a collection of 400 nodes in a 2500 square meter area. Frequency of the SOI is 157 MHz.

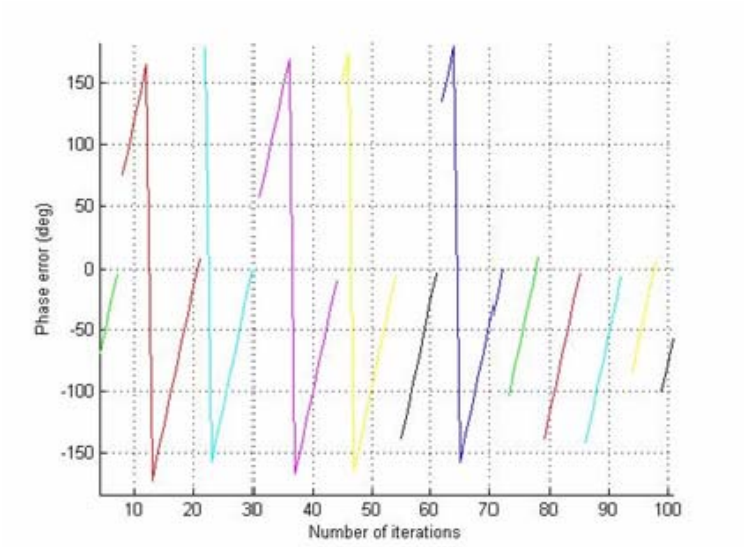


Figure 44. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 157 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 250000 square meter area.

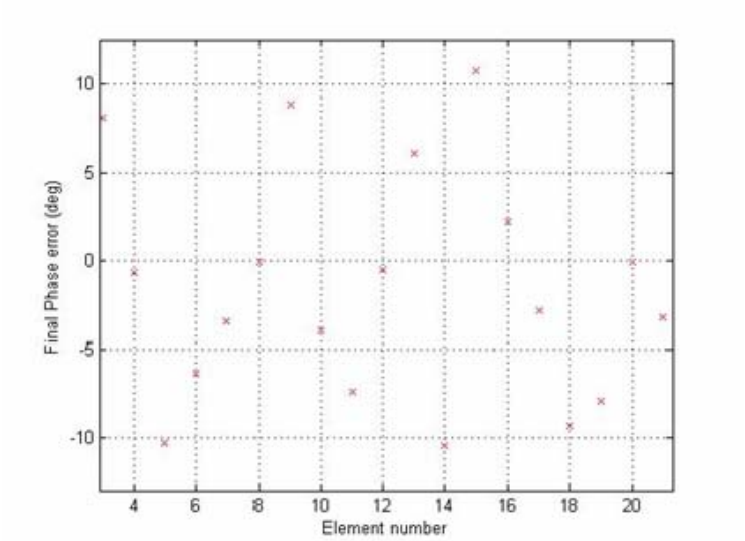


Figure 45. Final phase error for a sample of nodes from a collection of 400 nodes in a 250000 square meter area. Frequency of the SOI is 157 MHz.

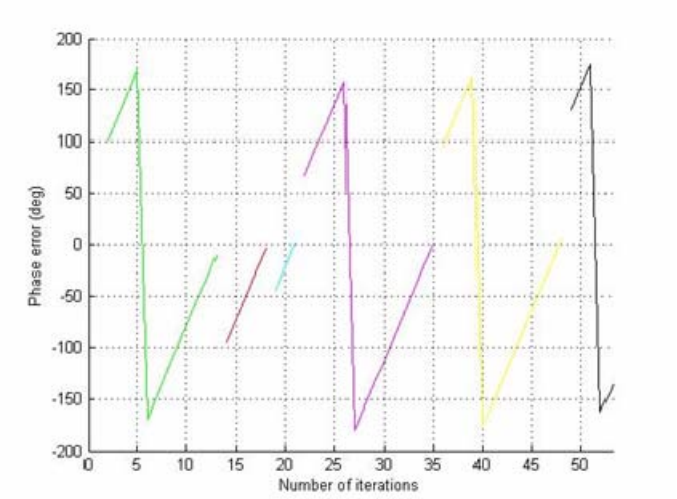


Figure 46. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 800 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 25 square meter area.

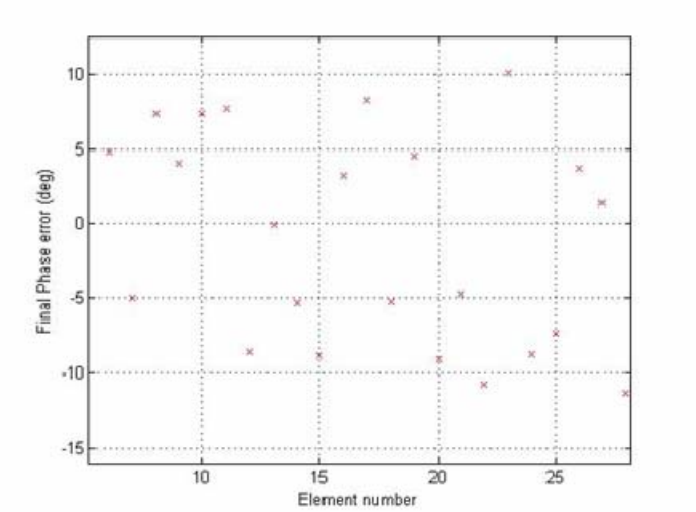


Figure 47. Final phase error for a sample of nodes from a collection of 400 nodes in a 25 square meter area. Frequency of the SOI is 800 MHz.

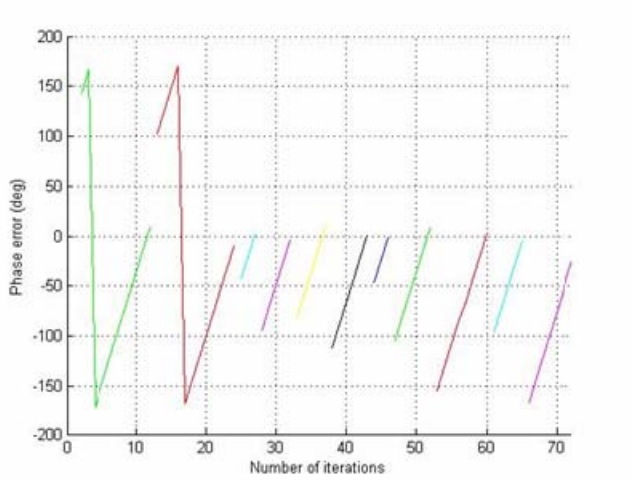


Figure 48. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 800 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 2500 square meter area.

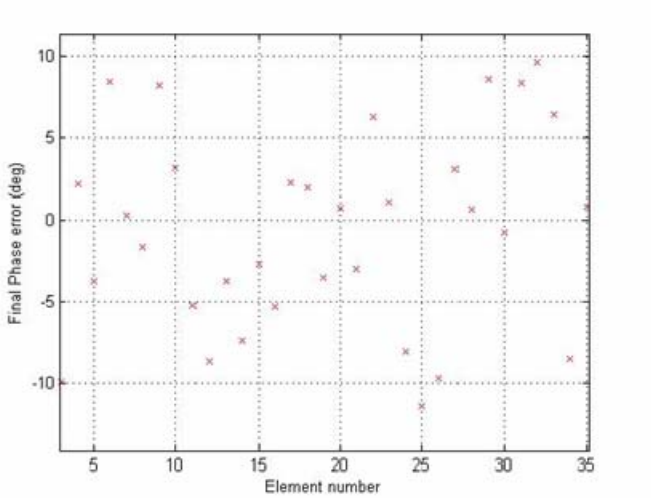


Figure 49. Final phase error for a sample of nodes from a collection of 400 nodes in a 2500 square meter area. Frequency of the SOI is 800 MHz.

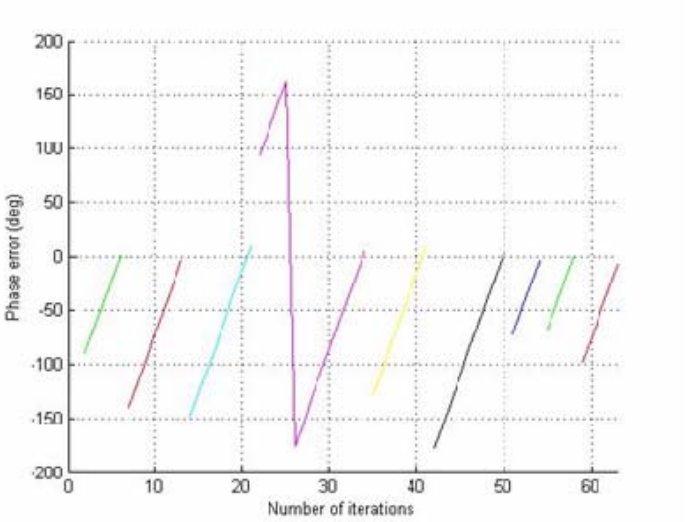


Figure 50. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 800 MHz. Shown is a highlight of nodes from a collection of 400 nodes in a 250000 square meter area.

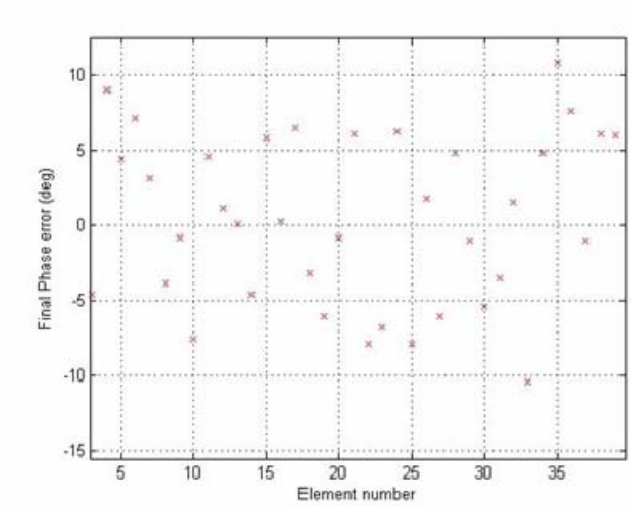


Figure 51. Final phase error for a sample of nodes from a collection of 400 nodes in a 250000 square meter area. Frequency of the SOI is 800 MHz.

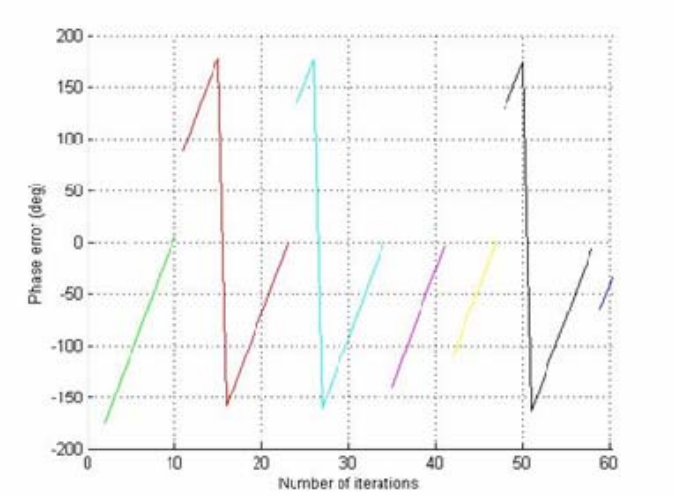


Figure 52. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 2.4 GHz. Shown is a highlight of nodes from a collection of 400 nodes in a 25 square meter area.

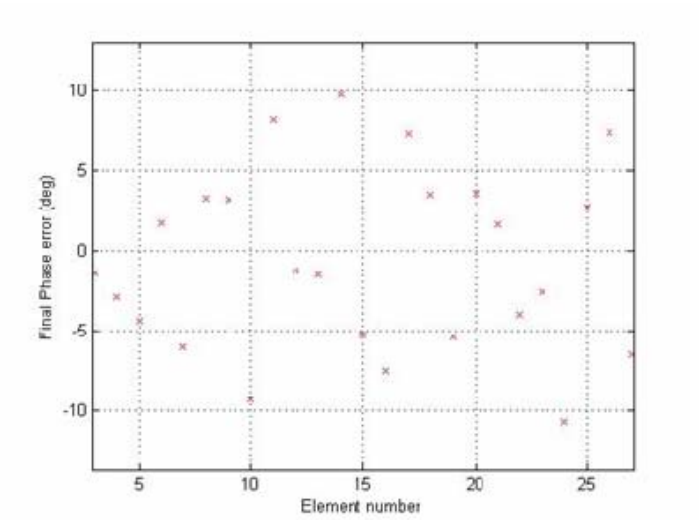


Figure 53. Final phase error for a sample of nodes from a collection of 400 nodes in a 25 square meter area. Frequency of the SOI is 2.4 GHz.

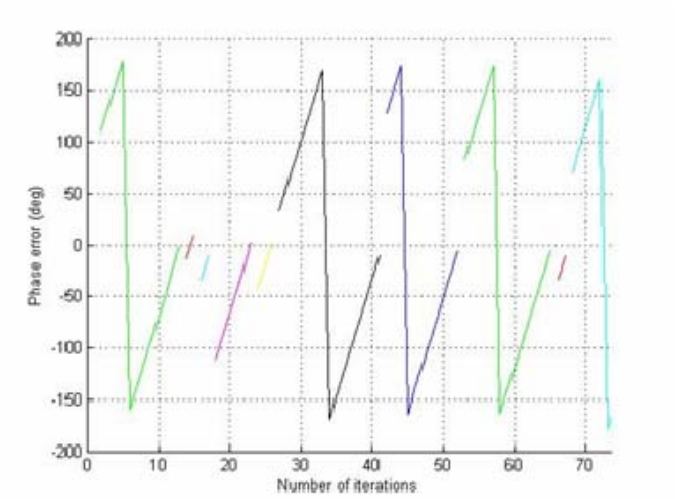


Figure 54. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 2.4 GHz. Shown is a highlight of nodes from a collection of 400 nodes in a 2500 square meter area.

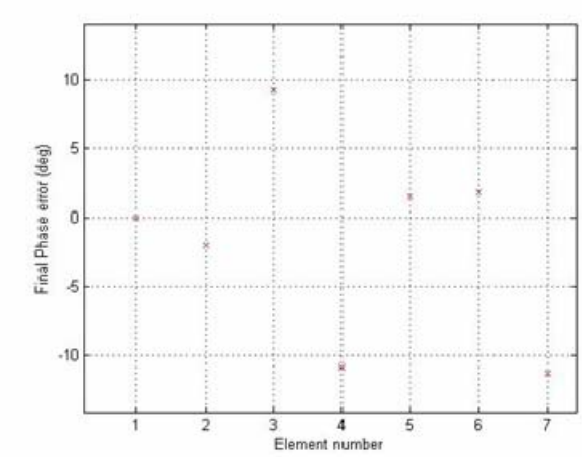


Figure 55. Final phase error for a sample of nodes from a collection of 400 nodes in a 2500 square meter area. Frequency of the SOI is 2.4 GHz.

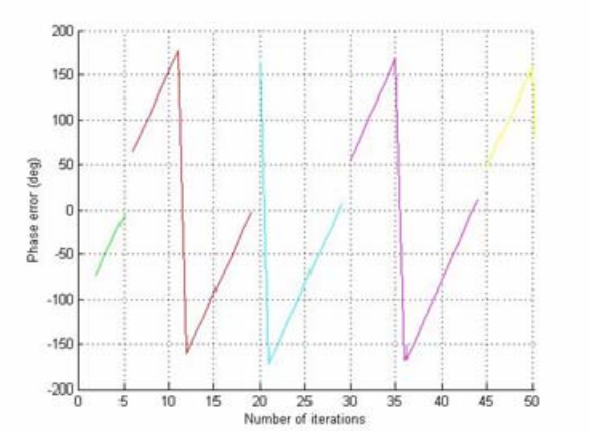


Figure 56. Number of iterations required for the Brute Force technique to phase synchronize at a frequency of 2.4 GHz. Shown is a highlight of nodes from a collection of 400 nodes in a 250000 square meter area.

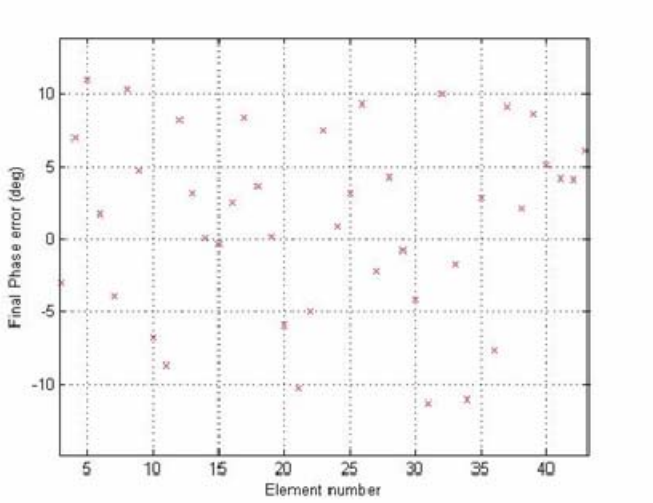


Figure 57. Phase error for a sample of nodes from a collection of 400 nodes in a 250000 square meter area. Frequency of the SOI is 2.4 GHz.

On average, the "brute force" technique requires half the total number of quantization levels to synchronize one element because the required phase shift is unknown.

2. "Beam Tagging" Synchronization Technique

The "beam tagging" method as proposed in [28] is for self-focusing or steering of an adaptive transmitting array. It is a technique of applying low-index phase modulation to one of two antennas aimed at the same target, and measuring resultant amplitude

modulation to correct the phase alignment between them. This technique has been used to phase-align lasers onto a target and for testing a large radar array [28]. The "beam tagging" technique can be implemented with more hardware modifications as was previously shown in Figure 10. As noted in [20], the key changes are the addition of a phase modulation circuit on the element synchronization block and an amplitude modulation (AM) receiver circuitry on the central controller. In each element synchronization block, the modem controller holds a phase shift command and is able, on special request, to modulate the phase rapidly by $\pm 90^{\circ}$ from the command phase.

The code used in [28] was again modified for the SIGINT/IW architecture. This code can be found in Appendix A & B. Simulation was performed to verify the "beam tagging" technique based on the same initial conditions used earlier. As discussed in reference [20], the phase corrections are performed until an opposite command is detected. This means that the element phase relative to reference has changed from lead to lag or vice versa and a balance condition is reached. The "beam tagging" technique detects this and terminates the synchronization cycle.

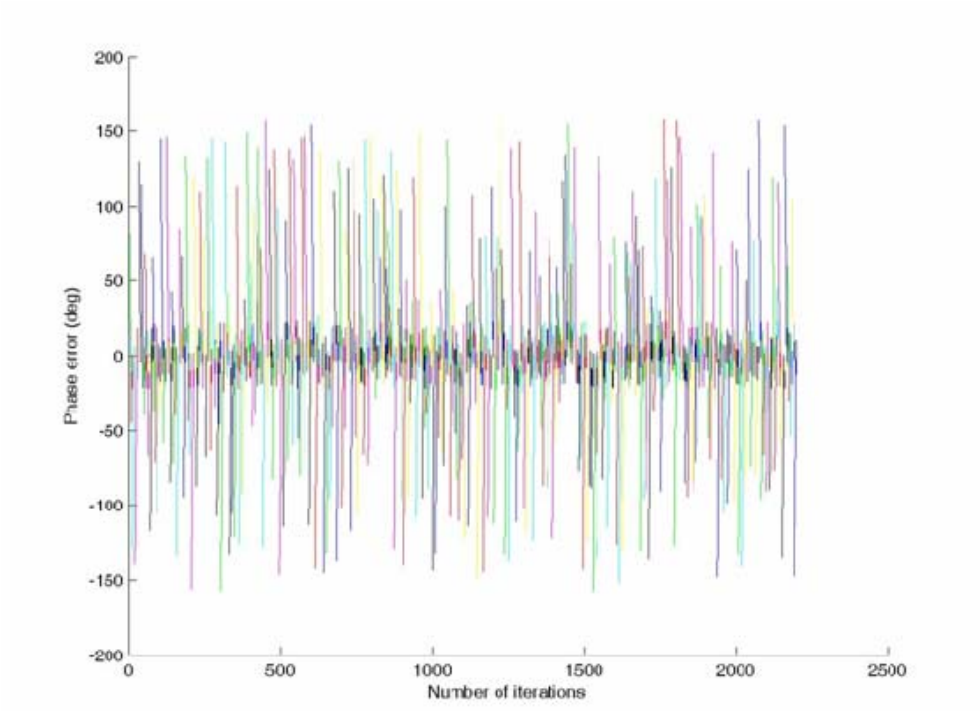


Figure 58. Synchronization of 400 Elements using the Beamtagging Technique.

The "beam tagging" technique synchronized the array after 2204 iterations, as shown in Figure 58, a significant reduction of *32%* in the number of iterations. The steady phase error is between $\pm 22.5^{\circ}$ as shown in Figure 59. The steady phase error is greater because when the synchronization cycle is terminated, the phase could have been overcorrected by 22.5°, giving rise to the error of $\pm 22.5^{\circ}$. These results are consistent with the results found in [28], which proves the viability of using either method to synchronize the individual antenna elements in a randomly distributed array.

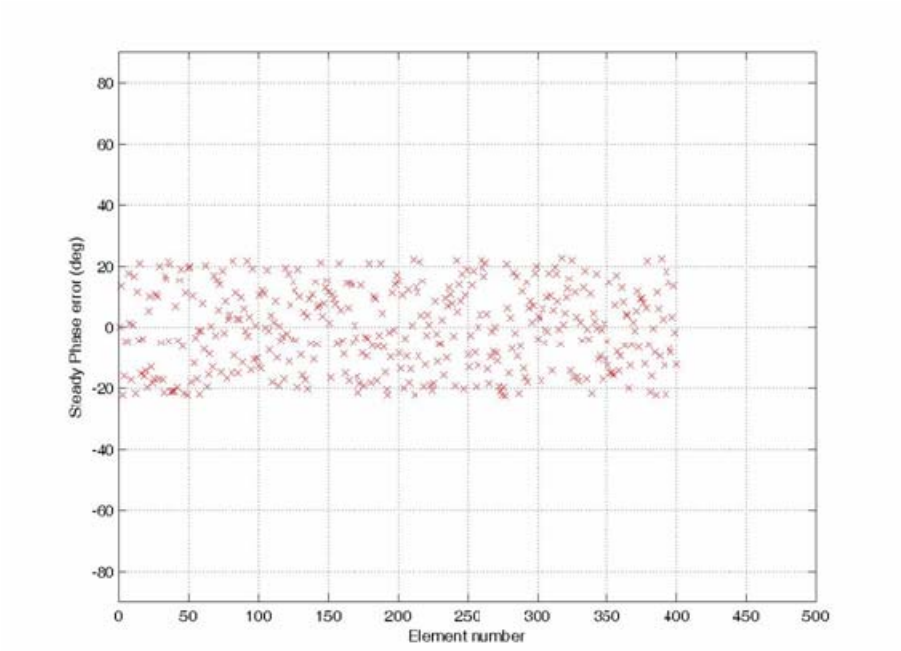


Figure 59. Phase Error of all Elements after Phase Synchronization using the "beam tag" technique.

In summary [28], concluded that even though the "beam tagging" technique was faster on the order of *2* to *3μs*, the more simple "brute force" technique is preferred for the wireless radar. Based on the research conducted for the SIGINT/IW antenna network using the modified MATLAB® algorithms the same conclusions can be drawn, that the brute force technique will be sufficient for the SIGINT/IW sensor architecture. Therefore, the SIGINT/IW sensor architecture will assume that synchronization will be conducted utilizing some variant of the "brute force" technique.

This chapter provided an overview into RF signals collection. A basic overview of antenna array theory and antenna array properties was discussed. Additionally, spectral estimation techniques were reviewed. Finally, the last section provided a look at the ongoing research by Jenn, *et al.* [20] into a difficult aspect of a wireless antenna. Even though synchronization is not central to this research it is important to give the reader an understanding of the complexity of this issue. Synchronization between the elements is assumed in this research. However, synchronization between the individual elements is a non-trivial requirement needed to make the SIGINT/IW wireless network a reality.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SUBSPACE METHODS

Most direction finding antennas are configured in a uniform manner, i.e. circular, linear, etc. The spacing between antenna elements is critical in the design of a traditional, uniformly spaced phased array. This is because when the spacing exceeds certain critical values, grating lobes will occur [3]. Grating lobes are additional sidelobes that have the same amplitude as the main beam. It is normally undesirable to have grating lobes, and in order to avoid them, the element spacing is required to meet the condition given by Equation (8). Equation (8) is rewritten as Equation (31) for convenience.

$$\frac{d}{\lambda_h} \leq \frac{1}{1+\left|\sin\theta_o\right|}, \tag{31}$$

where d is the element spacing, λ_h is the wavelength of the highest operating frequency and θ_0 is the scan angle.

Grating lobes require the coherent accumulation of the sampled signal at angles other than the steering angle. It will be shown that these grating lobes and the random distribution of our sensors cause a correlation between the signal and the noise that make traditional subspace methods, i.e. MUSIC, MLM, etc. non-optimal for solving for direction of arrival (DOA).

A. MULTIPLE SIGNAL CLASSIFICATION (MUSIC)

The MUltiple SIgnal Classification (MUSIC) algorithm introduced by Schmidt [47, 71] is a subspace based algorithm that exploits eigenstructure properties of the array correlation matrix. As discussed in references [30, 47, 69, 73], MUSIC can be used for estimating a range of signal parameters including the number of incident signals and their DOA. MUSIC is also extensively used in spectral estimation to estimate the frequency [73]. Earlier the spectral estimation properties of MUSIC were demonstrated. The angle of arrival determination properties will now be presented. Several variations of MUSIC exist, but in this research we will only investigate the standard form [30, 47, 69, 73].

As previously stated MUSIC separates the received signal plus noise into a signal subspace and a noise subspace using eigenvalue decomposition [16, 30, 69, 73]. Once the noise subspace has been estimated, a search for angle of arrival of the desired signal is made by looking for steering vectors that are as orthogonal to the noise subspace as possible [73]. The search may be performed using either the noise subspace or the signal subspace over all possible DOAs to determine the steering vectors. The steering vectors corresponding to the directional sources are orthogonal to the noise subspace and are found within the signal subspace [16, 30, 69, 73].

An example used in [71], illustrates this process. For an M element array with D signals incident on the array, the received input data vector may be expressed as a linear combination of the D incident waveforms and noise.

$$\begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} = \begin{bmatrix} \Psi(\Theta_1) & \cdots & \Psi(\Theta_D) \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_D \end{bmatrix} + \begin{bmatrix} n_1 \\ \vdots \\ n_D \end{bmatrix}, \tag{32}$$

$$\underline{b} = \Psi \underline{s} + \underline{n}, \tag{33}$$

where $\underline{s}$ is the vector of incident signals, $\underline{n}$ is the noise vector, and $\underline{\Psi}(\Theta_j)$ is the array factor or the array steering vector corresponding to the direction of arrival of the j^{th} signal. Viewing the received vector $\underline{b}$ and the steering vector $\underline{\Psi}(\Theta_j)$ as vectors in M dimensional space, it can be seen that $\underline{b}$ is a particular linear combination of the array steering vectors with s_1 through s_D being the coefficients of the combination. Using this data model, the input covariance matrix R_{BB} can be expressed as

$$R_{BB} = E\left[bb^H\right] = \Psi E\left[ss^H\right]\Psi^H + E\left[nn^H\right], \tag{34}$$

$$R_{BB} = E\left[bb^H\right] = \Psi E\left[ss^H\right]\Psi^H + \sigma^2{}_{noise} I, \tag{35}$$

where R_{SS} is the signal correlation matrix $E\left[ss^H\right]$.

Using eigenvalue decomposition R_{BB} is separated into a K-dimensional signal subspace and a $(M-K)$-dimensional noise subspace. The eigenvectors associated with the smaller eigenvalues of the correlation matrix correspond to the noise subspace, and

the eigenvectors associated with the larger eigenvalues correspond to the signal subspace. Estimating R_{BB} with a finite data sample can cause the eigenvalues corresponding to the noise power to not be identical. They will, however, appear as a tightly spaced cluster of eigenvalues with the variance decreasing as the number of data samples used to estimate R_{BB} increases. Once the multiplicity of the smallest eigenvalue is determined, the number of signals is given by [47, 69, 71]

$$D = M - K \,, \tag{36}$$

where M is the number of elements and K is the number of eigenvalues corresponding to the noise subspace. Based on the definition of eigenvalues and eigenvectors, the eigenvectors corresponding to the smallest eigenvalues satisfy

$$R_{BB}\upsilon_i = \sigma^2{}_{noise}\upsilon_i, \quad \text{for } i = D+1,\ldots,M \,, \tag{37}$$

Observing Equation (35), we see that

$$\Psi R_{ss}\Psi^H\upsilon_i = 0, \quad \text{for } i = D+1,\ldots,M \,, \tag{38}$$

Since Ψ is full column rank and R_{SS} is non-singular, it follows that

$$\Psi^H\upsilon_i = 0, \quad \text{for } i = D+1,\ldots,M \,, \tag{39}$$

This implies that the column vectors of Ψ are perpendicular to the eigenvectors $\upsilon_{D+1},\ldots,\upsilon_M$ corresponding to the noise.

Summarizing the above example, the MUSIC method is based on the observation that the steering vectors corresponding to the DOA of the signals are orthogonal to the noise [16, 30, 47, 58, 62, 66, 69, 71]. Once the noise subspace as been calculated from the estimated array correlation matrix in Equation (35), then Ψ can then be searched to find the eigenvectors that are orthogonal to the noise subspace. The estimates of the DOAs are taken as those φ that give the smallest value of $\underline{\Psi}^H(\phi)E_{noise}$, i.e., the values that result in a steering vector furthest away from the noise subspace [16, 30, 47, 58, 62, 66, 69, 71]. The MUSIC spectrum can be expressed as

$$P_{MUSIC}(\phi) = \frac{1}{\underline{\Psi}^H(\phi)E_{Noise}E_{Noise}{}^H\underline{\Psi}(\phi)} \,, \tag{40}$$

The matrix $E_{Noise}E_{Noise}{}^H$ is a projection matrix onto the noise subspace. For steering vectors that are orthogonal to the noise subspace, the denominator of Equation

(40) will become very small, and thus the peaks will occur in $P_{MUSIC}(\phi)$ corresponding to the angle of arrival of the signal [16, 30, 47, 58, 62, 66, 69, 71]. When the ensemble average of the array input is known and the noise can be considered uncorrelated and identically distributed between the elements, the peaks of the MUSIC spectrum are guaranteed to correspond to the angle-of-arrival of the target signals [16, 30, 47, 58, 62, 66, 69, 71]. However, because of the grating lobes discussed earlier the signal and noise are not uncorrelated between elements. Based on the earlier discussion in Chapter IV, Section A, Subsection 1, about grating lobes it will be shown that MUSIC is non-optimal for the SIGINT/IW array [16, 30, 47, 58, 62, 66, 69, 71].

As can be seen in Figures 60 – 71, MUSIC is very sensitive to element spacing. This sensitivity will be illustrated by utilizing a variant of Equation (40) called Root MUSIC, as shown in Equation (41) to calculate the optimal spacing. This method is only applicable when using a ULA, but it provides a means to calculate the optimal spacing. The optimal spacing, *d*, for angle-of-arrival of 30° was calculated to be 0.1591549 λ. In all of the frequencies investigated, the calculated angle-of-arrival was determined to be to the right of the real azimuth when the spacing was less than the optimal. Likewise, the calculated angle-of-arrival was determined to be to the left of the real azimuth when the spacing was greater than *d*. The information used for the simulations is summarized in Tables 3-11. Tables 3-5 are for frequency 2.4 GHz. Tables 6-8 are frequency 800 MHz and finally Tables 9-11 are for frequency 157 MHz.

$$P_{MUSIC}(\phi) = \frac{1}{\sum_{i=M+1}^{N} E_i E_i^*}, \tag{41}$$

Number of Elements	Distance (m) λ = 0.125 meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
4	0.15λ	2400	30.295°	27.4219°
4	0.17λ	2400	30.136°	31.6406°
4	0.2λ	2400	29.991°	36.5625°

Table 3. A four element ULA with a test frequency of 2.4 GHz at three different element spacings.

Number of Elements	Distance (m) $\lambda = 0.125$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
10	0.15λ	2400	30.034°	27.4219°
10	0.17λ	2400	29.993°	31.6406°
10	0.2λ	2400	29.994°	36.5625°

Table 4. A ten element ULA with a test frequency of 2.4 GHz at three different element spacings.

Number of Elements	Distance (m) $\lambda = 0.125$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
20	0.15λ	2400	29.998°	27.4219°
20	0.17λ	2400	30.018°	31.6406°
20	0.2λ	2400	29.991°	36.5625°

Table 5. A twenty element ULA with a test frequency of 2.4 GHz at three different element spacings.

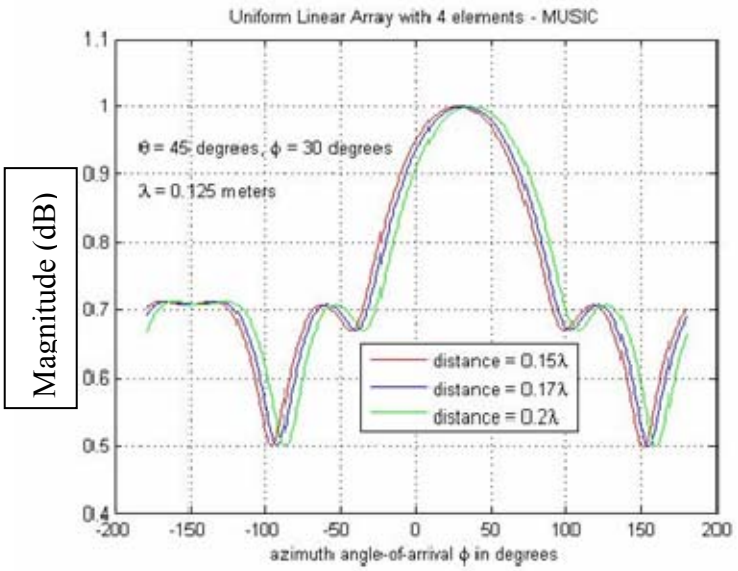


Figure 60. 2-D Array Pattern for a 4 element ULA at 2.4 GHz utilizing MUSIC.

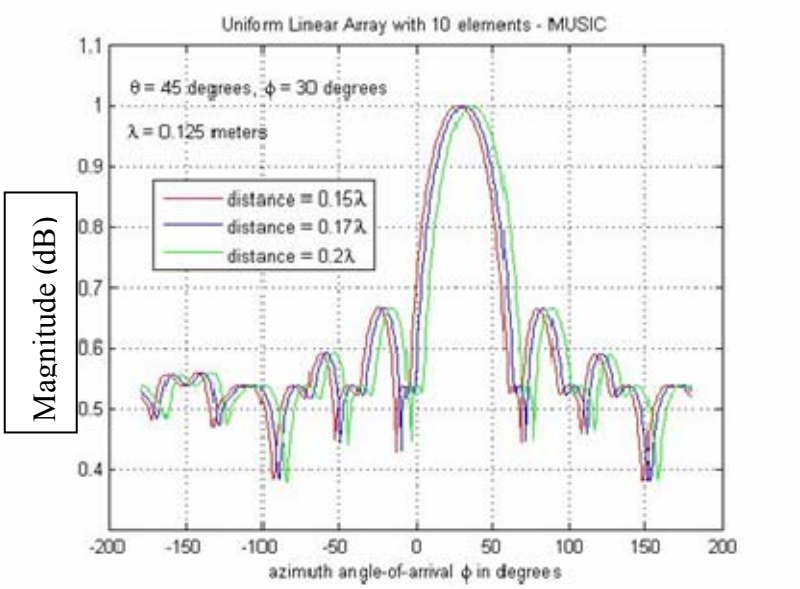


Figure 61. 2-D Array Pattern for a 10 element ULA at 2.4 GHz utilizing MUSIC.

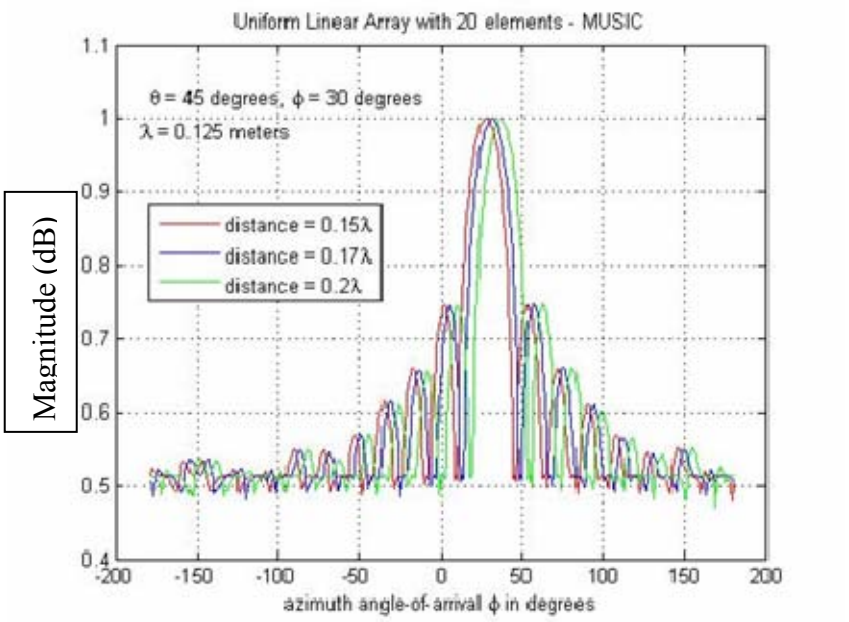


Figure 62. 2-D Array Pattern for a 20 element ULA at 2.4 GHz utilizing MUSIC.

Number of Elements	Distance (m) $\lambda = 0.375$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
4	0.15λ	800	29.562°	27.4219°
4	0.17λ	800	29.936°	30.9375°
4	0.2λ	800	30.448°	36.5625°

Table 6. A four element ULA with a test frequency of 800 MHz at three different element spacings. MHz.

Number of Elements	Distance (m) $\lambda = 0.375$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
10	0.15λ	800	29.975°	27.4219°
10	0.17λ	800	30.1°	31.6406°
10	0.2λ	800	30.06°	36.5625°

Table 7. A ten element ULA with a test frequency of 800 MHz at three different element spacings.

Number of Elements	Distance (m) $\lambda = 0.375$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
20	0.15λ	800	29.989°	27.4219°
20	0.17λ	800	30.015°	31.6406°
20	0.2λ	800	29.993°	36.5625°

Table 8. A twenty element ULA with a test frequency of 800 MHz at three different element spacings.

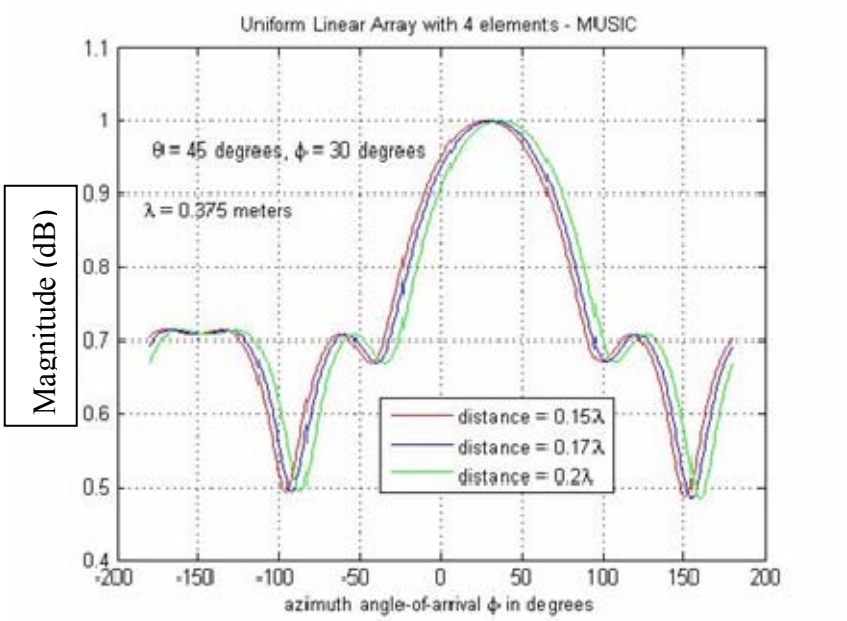


Figure 63. 2-D Array Pattern for a 4 element ULA at 800 MHz utilizing MUSIC.

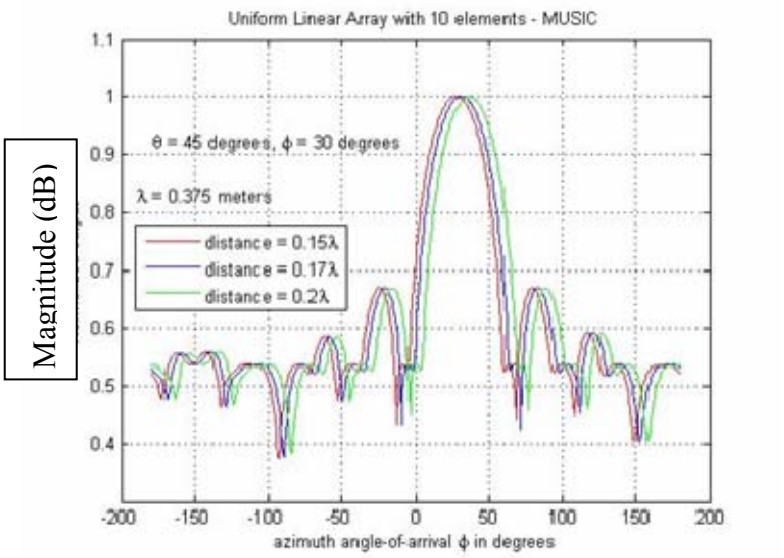


Figure 64. 2-D Array Pattern for a 10 element ULA at 800 MHz utilizing MUSIC.

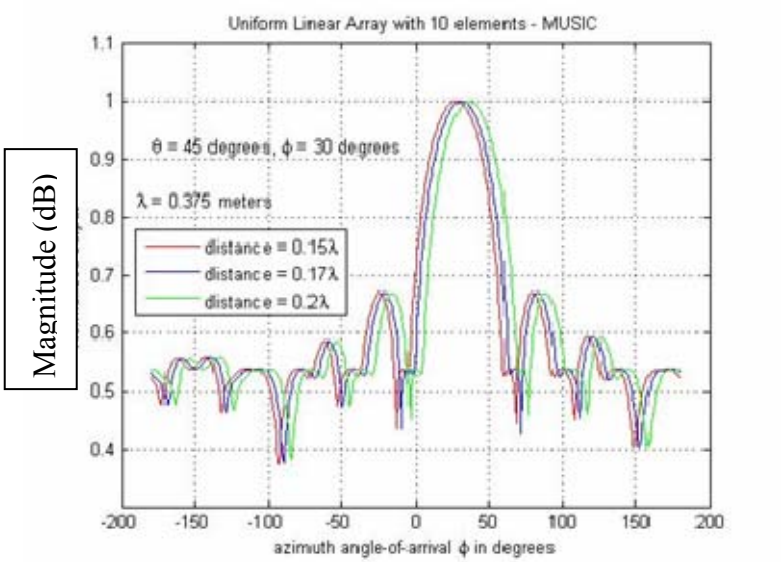


Figure 65. 2-D Array Pattern for a 20 element ULA at 800 MHz utilizing MUSIC.

Number of Elements	Distance (m) $\lambda = 2$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
4	0.15λ	157	29.896°	27.4219°
4	0.17λ	157	29.749°	31.6406°
4	0.2λ	157	29.943°	36.5625°

Table 9. A four element ULA with a test frequency of 157 MHz at three different element spacings.

Number of Elements	Distance (m) $\lambda = 2$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
10	0.15λ	157	30.04°	27.4219°
10	0.17λ	157	30.045°	30.9375°
10	0.2λ	157	30.0024°	36.5625°

Table 10. A ten element ULA with a test frequency of 157 MHz at three different element spacings.

Number of Elements	Distance (m) $\lambda = 2$ meters	Frequency (MHz)	DOA RootMUSIC	Azimuth Calculated
20	0.15λ	157	30.004°	27.4219°
20	0.17λ	157	29.968°	30.9375°
20	0.2λ	157	29.992°	36.5625°

Table 11. A twenty element ULA with a test frequency of 157 MHz at three different element spacings.

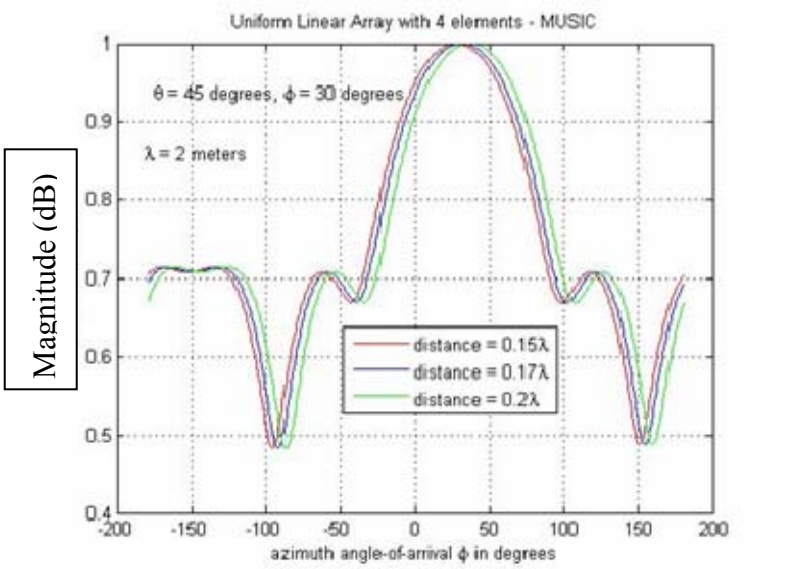


Figure 66. 2-D Array Pattern for a 4 element ULA at 157 MHz utilizing MUSIC.

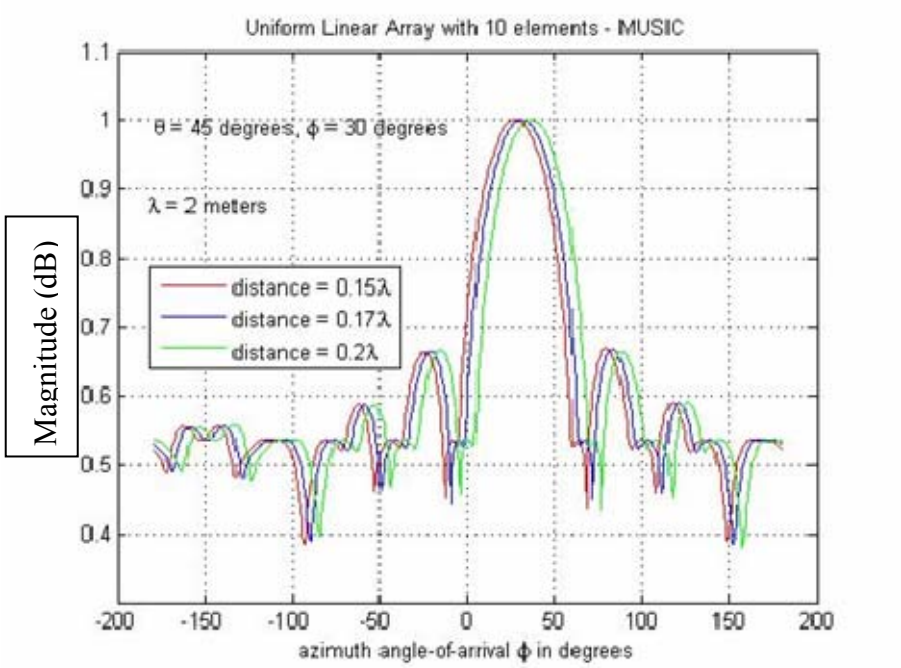


Figure 67. 2-D Array Pattern for a 10 element ULA at 157 MHz utilizing MUSIC.

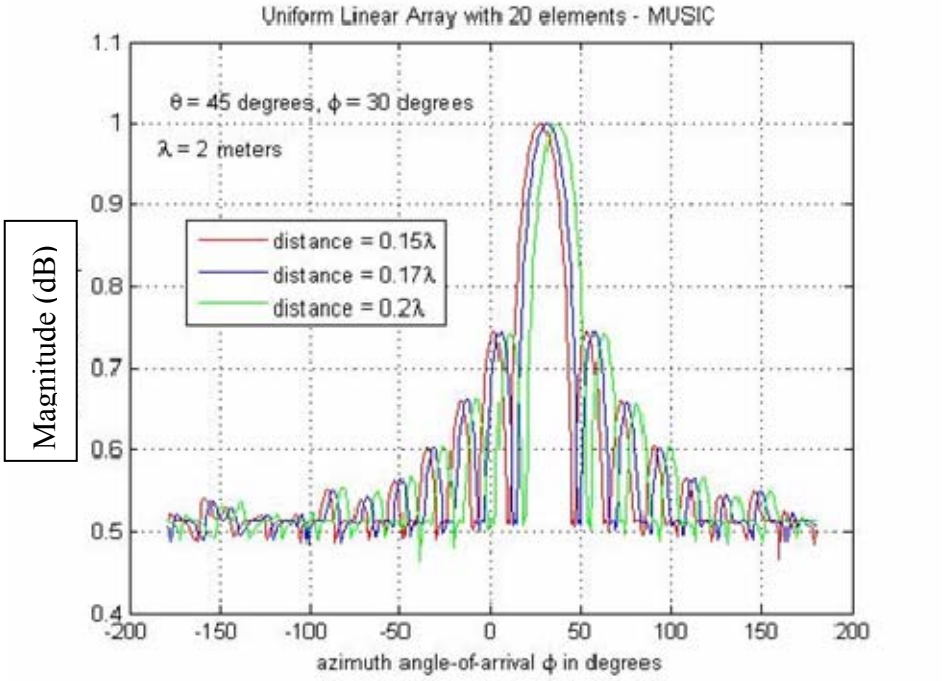


Figure 68. 2-D Array Pattern for a 20 element ULA at 157 MHz utilizing MUSIC.

The results from the above simulations were taken at frequencies of 157 MHz, 800 MHz and 2.4 GHz. The antenna element spacings were normalized based on

wavelength. The largest element spacing distance was 0.4 meters apart for the 157 MHz frequency. The closest spacing was 0.01875 meters for the 2.4 GHz frequency. These spacings are very unrealistic for an antenna field that is expected to be deployed from artillery shells, dropped from UAVs or scattered via a UUV. Now a look at the array pattern characteristics when the elements are spaced at wavelengths equal to or greater than half-wavelength were investigated. The data for these simulations can be seen in Tables 12 – 14. Figures 69 – 71 illustrate that the inconsistency in the array pattern is much worse as the element spacings increase.

Number of Elements	Distance	Frequency
40	$\frac{\lambda}{2}$	2.4 GHz
40	$\frac{3\lambda}{2}$	2.4 GHz
40	2λ	2.4 GHz

Table 12. A forty element ULA with a test frequency of 2.4 GHz at three different element spacings equal to or greater than half-wavelength.

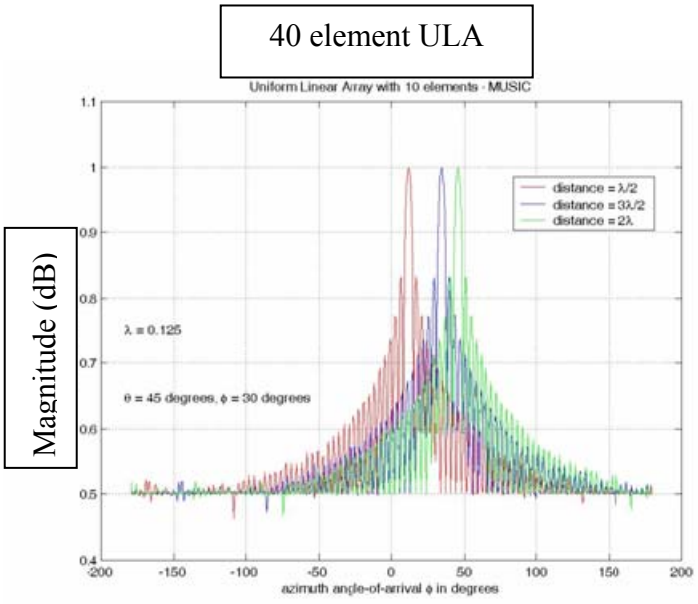


Figure 69. 2-D Array Pattern for a 40 element ULA at 2.4 GHz utilizing MUSIC where the elements are spaced at half-wavelength or greater.

Number of Elements	Distance	Frequency
40	$\frac{\lambda}{2}$	800 MHz
40	$\frac{3\lambda}{2}$	800 MHz
40	2λ	800 MHz

Table 13. A forty element ULA with a test frequency of 800 MHz at three different element spacings equal to or greater than half-wavelength.

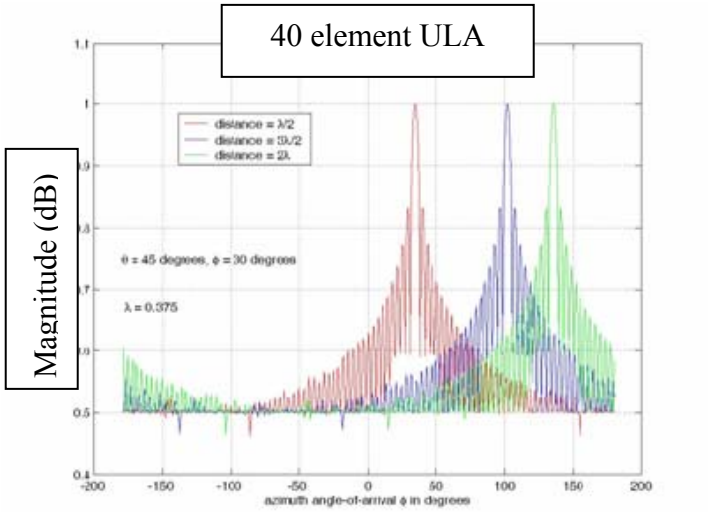


Figure 70. 2-D Array Pattern for a 40 element ULA at 800 MHz utilizing MUSIC where the elements are spaced at half-wavelength or greater.

Number of Elements	Distance	Frequency
40	$\frac{\lambda}{2}$	157 MHz
40	$\frac{3\lambda}{2}$	157 MHz
40	2λ	157 MHz

Table 14. A forty element ULA with a test frequency of 157 MHz at three different element spacings equal to or greater than half-wavelength.

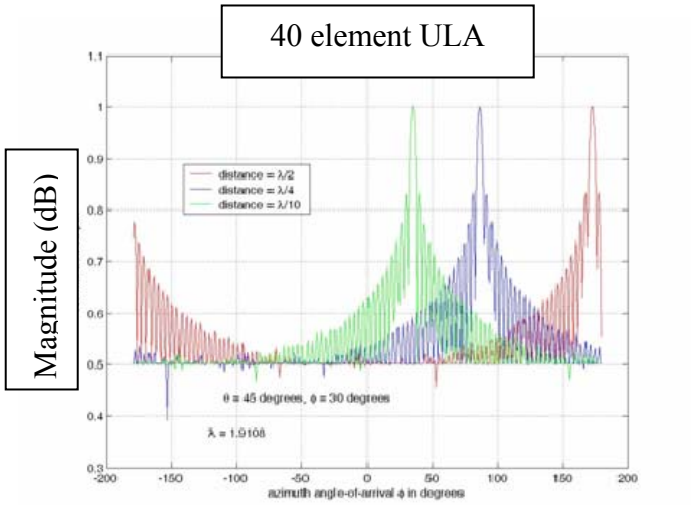


Figure 71. 2-D Array Pattern for a 40 element ULA at 157 MHz utilizing MUSIC where the elements are spaced at half-wavelength or greater.

When more realistic distances are used it can be seen in the above figures how the results become very erratic. Other variants and/or adaptations of MUSIC may be able to better cope with this phenomena but these will not be explored in this research. Now we'll take a look into the Maximum Likelihood method.

B. MAXIMUM LIKELIHOOD

The method of Maximum Likelihood (ML) estimation is one of the most popular estimation methods used in signal processing. For a DOA estimation problem, the term "maximum likelihood" is used for the method that finds the ML estimate of the direction rather than of the power [16, 33]. This method uses the array weights, which are obtained by minimizing the mean output power in the look direction [16, 30, 47, 58, 62, 66, 69, 71].

An expression for the power spectrum is given by

$$P_{MV}(\phi) = \frac{1}{\underline{\Psi}^{H}(\phi) R_{BB}^{-1} \underline{\Psi}(\phi)}, \tag{42}$$

Just like with MUSIC due to the grating lobes discussed earlier in Chapter IV, Section A, Subsection 1, MLM is non-optimal for the SIGINT/IW array. This can be easily seen in Figures 72 – 83.

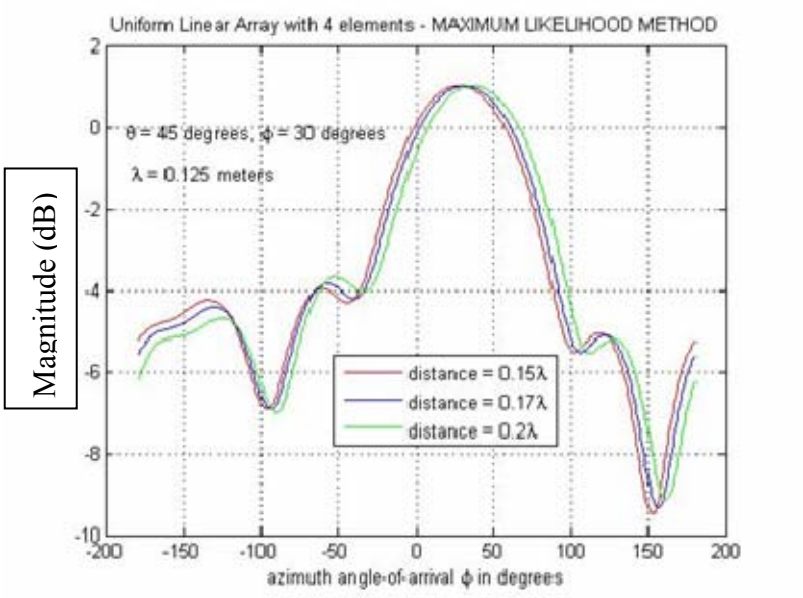


Figure 72. 2-D Array Pattern for a 4 element ULA at 2.4 GHz utilizing MLM.

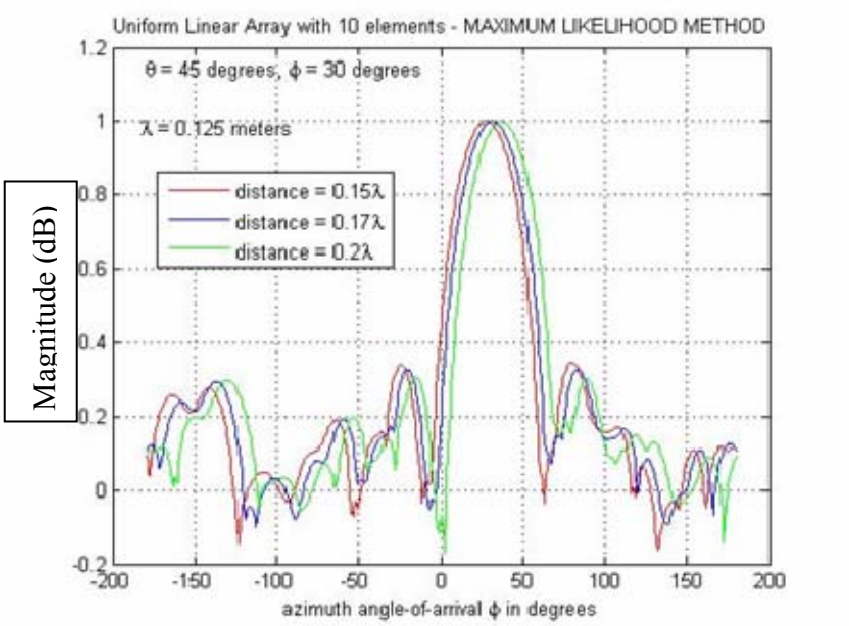


Figure 73. 2-D Array Pattern for a 10 element ULA at 2.4 GHz utilizing MLM.

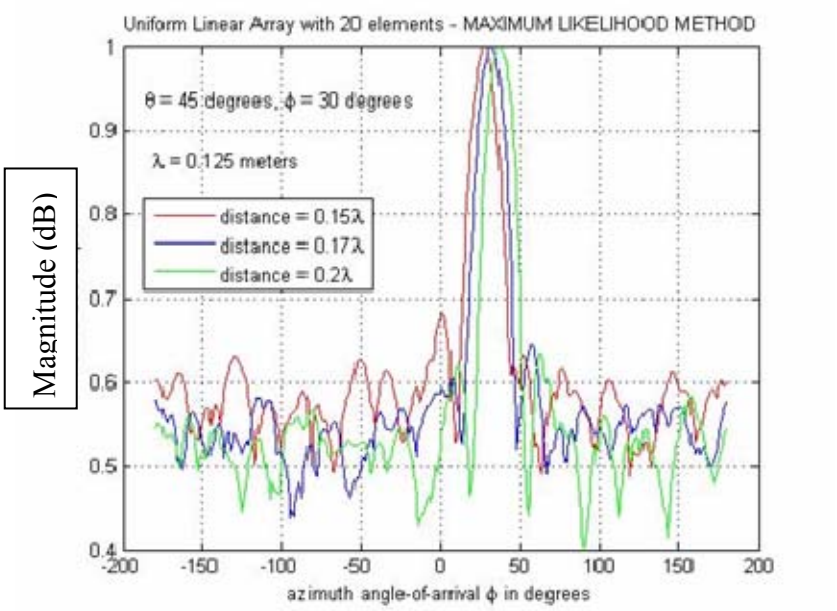


Figure 74. 2-D Array Pattern for a 20 element ULA at 2.4 GHz utilizing MLM.

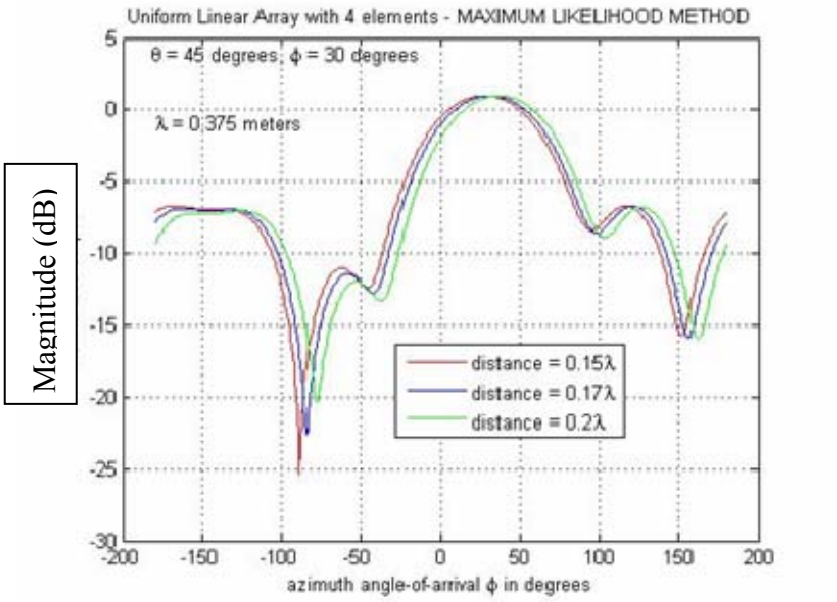


Figure 75. 2-D Array Pattern for a 4 element ULA at 800 MHz utilizing MLM.

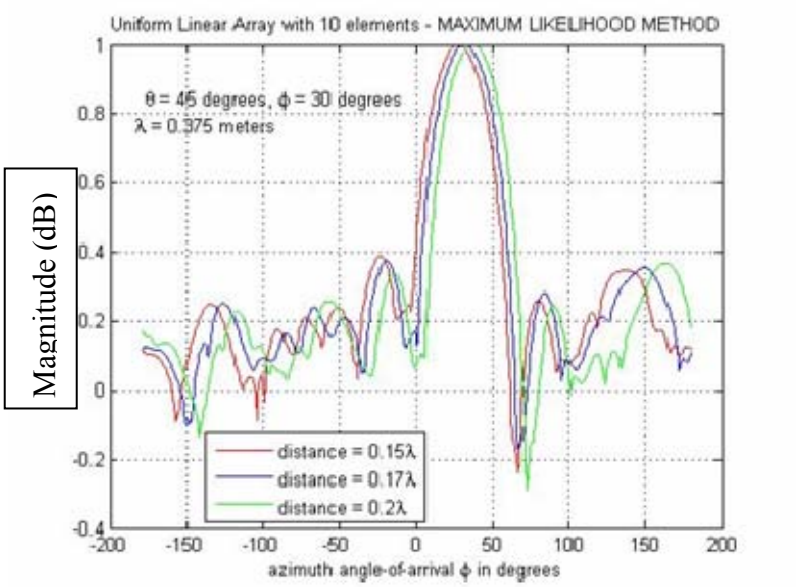


Figure 76. 2-D Array Pattern for a 10 element ULA at 800 MHz utilizing MLM.

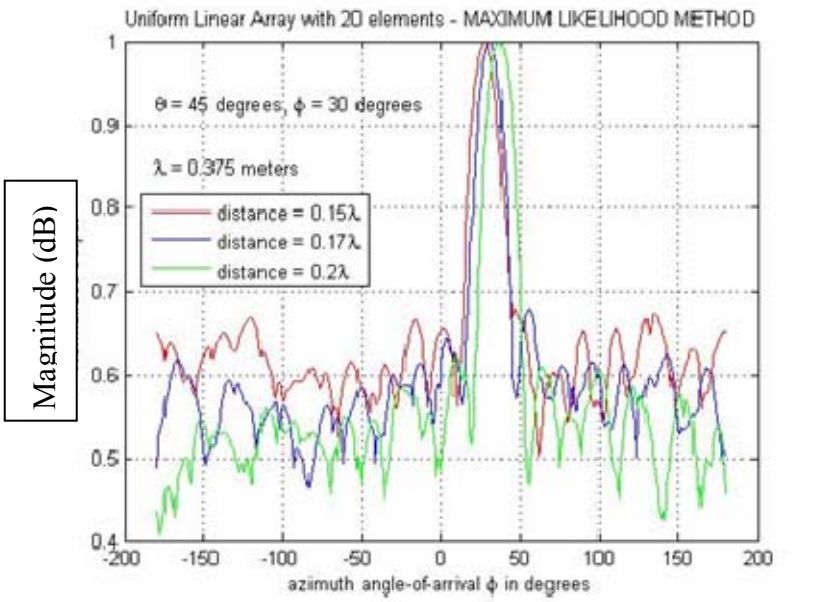


Figure 77. 2-D Array Pattern for a 20 element ULA at 800 MHz utilizing MLM.

As in the case for MUSIC, the results from these simulations were taken at frequencies of 157 MHz, 800 MHz and 2.4 GHz. The antenna elements were spaced at fractions of wavelengths apart for convenience of computation. The largest spacing was 0.4 meters apart for the 157 MHz frequency. The closest spacing was 0.01875 meters for the 2.4 GHz frequency.

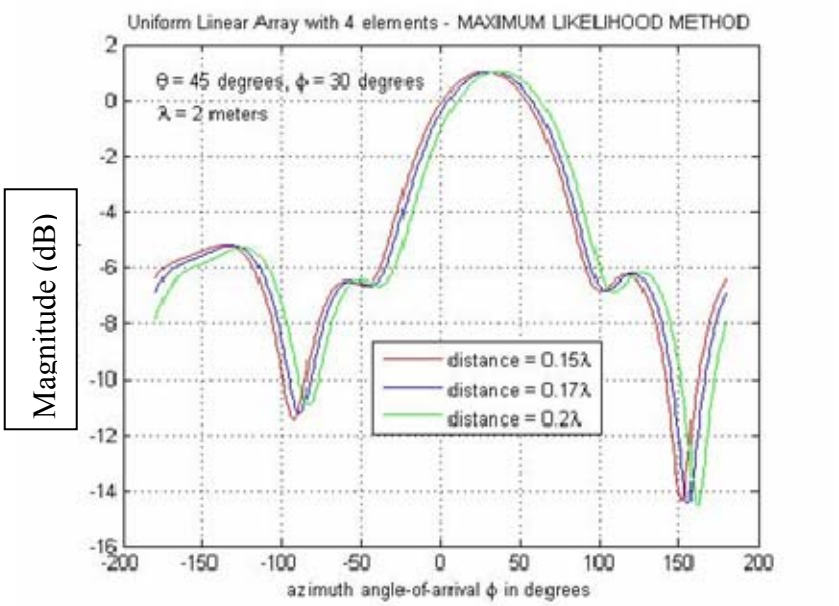


Figure 78. 2-D Array Pattern for a 4 element ULA at 157 MHz utilizing MLM.

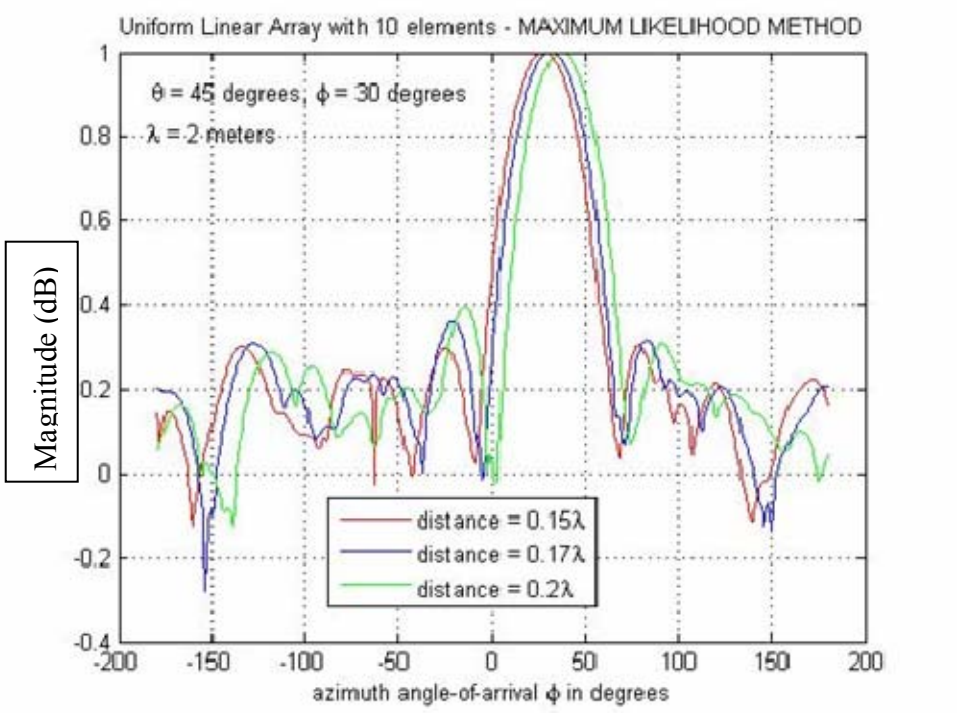


Figure 79. 2-D Array Pattern for a 10 element ULA at 157 MHz utilizing MLM.

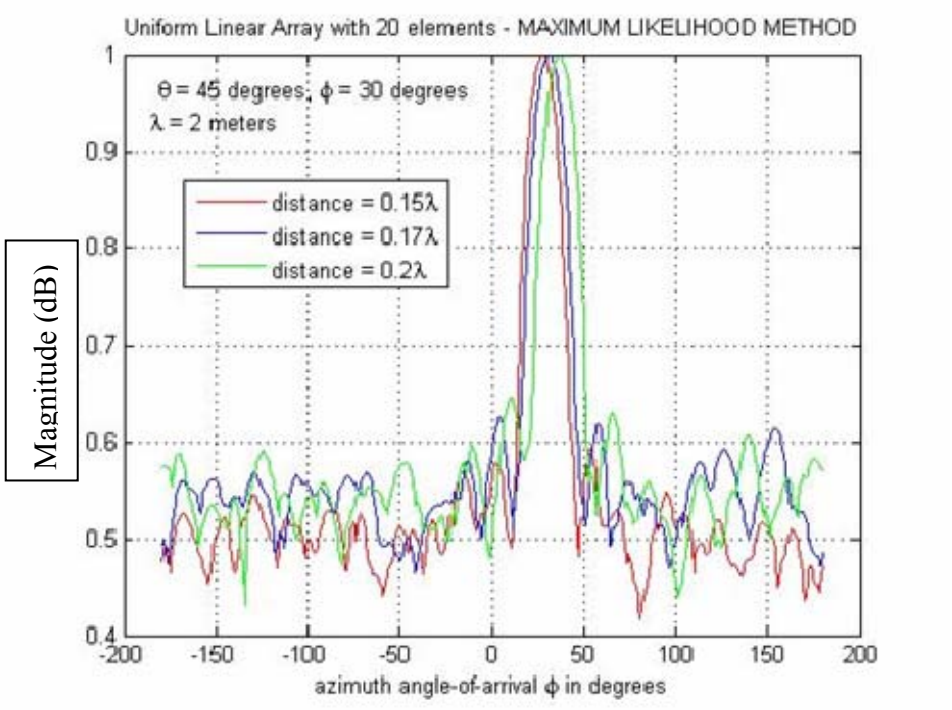


Figure 80. 2-D Array Pattern for a 20 element ULA at 157 MHz utilizing MLM.

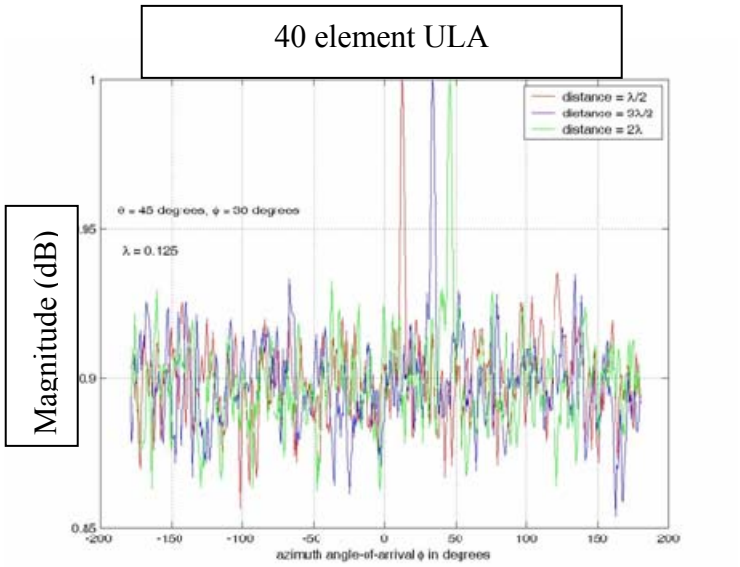


Figure 81. 2-D Array Pattern for a 40 element ULA at 2.4 GHz utilizing MLM.

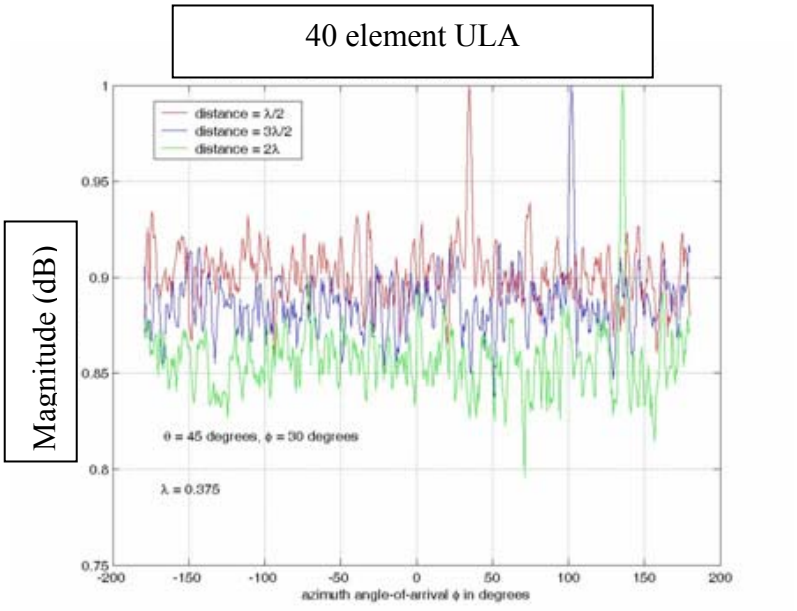


Figure 82. 2-D Array Pattern for a 40 element ULA at 800 MHz utilizing MLM.

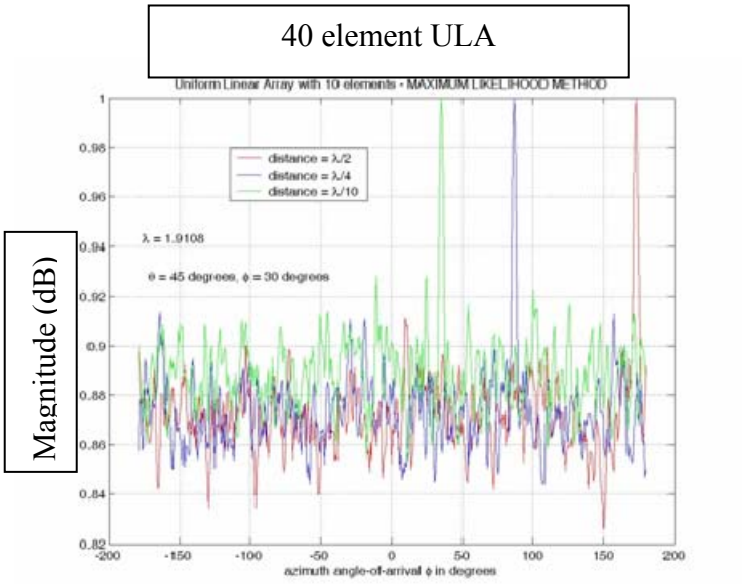


Figure 83. 2-D Array Pattern for a 40 element ULA at 157 MHz utilizing MLM.

As in the case of MUSIC, when the elements become further apart the results become more erratic. Other variants and/or adaptations of MLM may be able to better cope with this phenomena but these will not be explored in this research. Therefore, it

will be impractical to impossible to use traditional subspace methods to determine the angle of arrival. Hence, this research will investigate methods that can mitigate the grating lobe problem.

This chapter has provided insight into the problems with using traditional subspace methods with the SIGINT/IW wireless sensor array. Since the sensor nodes will in some cases be spaced greater than a half-wavelength apart, the grating lobes cause the signal and noise to become correlated rendering the traditional subspace methods inadequate for determining the direction of arrival. Therefore this leads into the next phase of the research, which is to find a method suitable for determining this direction of arrival in order to utilize beamforming to enhance the collection against the target.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. DIRECTION-OF-ARRIVAL

This chapter provides a background into the two methods analysed in this dissertation for line of bearing determination, time difference of arrival and beamforming. Both methods were analyzed in order to determine the most expeditious and energy efficient method to determine a line of bearing to the target. This chapter will give the reader an appreciation into some of the constraints and tradeoffs associated with both methods.

A. TIME DISTANCE-OF-ARRIVAL (TDOA)

Time difference of arrival (TDOA) takes advantage of the fact that a transmitted signal will arrive at the different elements at different time instants. It will be shown that TDOA is not affected by the grating lobes, which bemoans the subspace methods of MUSIC and MLM. In fact it will be shown that as the baseline d, between the sensor node and the central controller increases the resulting TDOA measurement variance gets smaller and subsequently the resulting TDOA gets better. In the SIGINT/IW wireless antenna array each sensor forms a two-node collector pair with the central controller, as shown in Figure 84. The basic principle of the two-element linear array will then be extended to encompass the entire antenna grid. The sensors within our grid will take advantage of the fact that our grid is densely populated and that the sensors are spread over a wide area [29].

Defining TDOA in terms of the emitter and antenna element positions is accomplished using the vectors $\underline{r}_1$ and $\underline{r}_2$ from Figure 84 where $\underline{r}_1$ is the distance from the central controller to the emitter and $\underline{r}_2$ is the distance from an individual antenna element to the emitter [29].

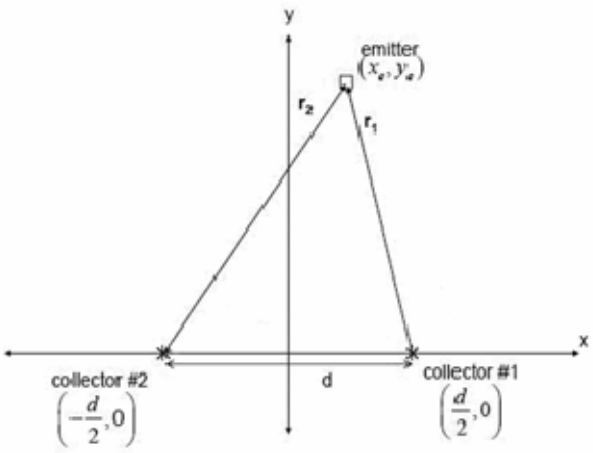


Figure 84. TDOA Geometry for a collector pair. [After Ref. 47]

Equation (44), illustrates the difference in the length of the two vectors $\underline{r}_1 = \left[x_e - d/2; y_e\right]^T$ and $\underline{r}_2 = \left[x_e + d/2; y_e\right]^T$, divided by the speed of light c, yielding the TDOA:

$$TDOA = \frac{|\underline{r}_2| - |\underline{r}_1|}{c} = \frac{1}{c}\left(\sqrt{\left(x_e + \frac{d}{2}\right)^2 + {y_e}^2} - \sqrt{\left(x_e - \frac{d}{2}\right)^2 + {y_e}^2}\right), \tag{43}$$

where x_e and y_e are the respective *x*-*y* coordinates of the emitter and d is the distance between the two-node collector pair commonly referred to as the baseline.

When the two-node collector pair are in motion, multiple TDOA measurements can be used to estimate emitter position in a manner analogous to a navigator performing triangulation. For the wireless antenna array network the two-node collector pairs are not in motion therefore only one measurement was taken from each two-node collector pair.

The narrowband signal shown in Equation (Q-10) will be assumed to be the signal-of-interest arriving at our two different sensors. The likelihood function for the received signal is shown in Equation (44) and is derived in detail in Appendix Q [72].

$$E\left[\tilde{r}(t)\right] = AI_0\left(\frac{2Aq}{N_0}\right)\exp\left[\frac{-1}{2N_0}\int_{\frac{-T}{2}}^{\frac{T}{2}}\left|\tilde{r}(t)\right|^2 + A^2\tilde{s}_{r_0}{}^2(t)\,dt\right],$$
$$q = \frac{1}{2}\left|\int_{\frac{-T}{2}}^{\frac{T}{2}}\tilde{r}(t)\tilde{s}_{r_0}{}^*(t)\,dt\right| \qquad (44)$$

where A is the amplitude of the received signal, $\tilde{r}(t)$ is the received signal's complex envelope plus noise as defined in Equation (Q-16). $\tilde{s}_{r_0}(t)$ is the complex amplitude of the received signal. N_0 is the noise power spectral density.

The signal is assumed to arrive in an interval between $-T/2$ and $T/2$. $I_0(\square)$ is the modified Bessel function order zero. This function is a result of factoring out the carrier phase as a nuisance parameter. The carrier phase is assumed to be a uniform random variable between $-\pi$ and π. An assumption is made that the SNR is high. The argument of the exponential is assumed to be constant and the only thing assumed to be unknown is the argument of the modified Bessel function. Therefore the likelihood function is maximized when the argument of the modified Bessel function, specifically, q is maximized [72].

It is assumed that the envelope of the received signal is a close approximation to the transmitted signal. Therefore, q can be expressed as

$$q = \frac{1}{2}\left|\int_{\frac{-T}{2}}^{\frac{T}{2}}\tilde{r}(t)\tilde{s}_t^*(t-\tau)\,dt\right| \qquad (45)$$

where the received signal has been rewritten in terms of the transmitted signal from Equation (6) only delayed in time, τ. As shown below

$$\tilde{s}_{r_0}{}^* = s_t(t-\tau) = A(t-\tau)e^{(j\gamma(t-\tau))}, \qquad (46)$$

The high frequency components have been removed during the filtering process. It is assumed that a quadrature detector with a matched filter is used. The time at which the envelope peaks is the approximate value of the delay [72].

The Cramér-Rao bound for the variance of this estimation taken from [72] will now be derived for the narrowband signal case. The received signal is assumed to be

$$\tilde{r}(t) = A\tilde{s}_t(t-\tau) + \tilde{n}(t), \tag{47}$$

The derivation uses the large signal approximation. Now taking the natural log of likelihood function results in the following:

$$\ln\left[p\left(\underset{\sim}{r} \mid \tau\right)\right] \cong \ln(A) + \ln\left(I_0\left(\frac{2Aq}{N_0}\right)\right) + \ln\left(\exp\left[\frac{-1}{2N_0}\int_{\frac{-T}{2}}^{\frac{T}{2}} \left|\tilde{r}(t)\right|^2 + A^2\tilde{s}_{r_0}{}^2(t)\,dt\right]\right), \tag{48}$$

when the argument of the modified Bessel function is maximum then

$$\ln I_0(x) \cong x\,. \tag{49}$$

Based on this assumption Equation (48) can be rewritten in the form

$$\ln\left[p\left(\underset{\sim}{r} \mid \tau\right)\right] \cong \ln(A) + \frac{2Aq}{N_0} + \underbrace{\left[\frac{-1}{2N_0}\int_{\frac{-T}{2}}^{\frac{T}{2}} \left|\tilde{r}(t)\right|^2 + A^2\tilde{s}_{r_0}{}^2(t)\,dt\right]}_{\text{This term is approximately independent of } \tau.}, \tag{50}$$

Due to its independence on τ, the last term in Equation (51) can be removed. Substituting back in for q from Equation (47), Equation (51) now becomes

$$\begin{aligned}\ln\left[p\left(\underset{\sim}{r} \mid \tau\right)\right] &\cong \ln(A) + \frac{A}{N_0}\left|\int_{\frac{-T}{2}}^{\frac{T}{2}} \tilde{r}(t)\,s_t^*(t-\tau)\,dt\right| \\ &\cong \ln(A) + \frac{A}{N_0}\left|\int_{\frac{-T}{2}}^{\frac{T}{2}} \left[A\tilde{s}_t(t-\tau) + \tilde{n}(t)\right]s_t^*(t-\tau)\,dt\right|\end{aligned}, \tag{51}$$

As discussed in [72], because of the large SNR assumption the noise term is dropped and $\tau \approx 0$ in the last term. Therefore Equation (51) is rewritten as

$$\ln\left[p\left(\tilde{r} \mid \tau\right)\right] \cong \ln\left(A\right) + \frac{A}{N_0}\left| \int_{\frac{-T}{2}}^{\frac{T}{2}} A s_t\left(t-\tau\right) s_t^*\left(t\right) dt \right|, \tag{52}$$

An important aspect of calculating the time difference of arrival is the variance associated between multiple measurements. This variance determines the plus or minus error associated with the subsequent line of bearing determination. Equation (52) provides convenient representation for solving for the Cramér-Rao bound of the associated time difference measurement variance. Now a more detailed discussion of the cross-ambiguity function will be offered. The cross-ambiguity function will be utilized to calculate the time difference of arrival.

1. Cross-Ambiguity Function (CAF)

The complex envelope of the transmitted signal is collected by the sensors then the sampled signals are sent back to the central controller where the cross-ambiguity function (CAF) is used to determine the TDOAs. An FDOA, can also be determined from calculating the CAF, which could be very useful if prosecuting a mobile target [24]. However, our target is assumed stationary. The complex envelopes of the two signals will be denoted by $s_t\left(t-\tau\right)$ and $s_t^*\left(t\right)$, where t is the time of the first sample, and τ is the time lag between the two. The CAF is defined as

$$\begin{aligned} CAF\left(\tau, w\right) &= \frac{1}{2E_t} \int_{\frac{-T}{2}}^{\frac{T}{2}} \tilde{s}_t\left(t-\tau\right) \tilde{s}_t^*\left(t\right) e^{-j2\pi ft} dt \\ CAF\left(0\right) = 1 &= \frac{1}{2E_t} \int_{\frac{-T}{2}}^{\frac{T}{2}} \tilde{s}_t\left(t\right) \tilde{s}_t^*\left(t\right) dt \qquad , \\ E_t &= \frac{1}{2} \int_{\frac{-T}{2}}^{\frac{T}{2}} \left|\tilde{s}_t\left(t\right)\right|^2 dt \end{aligned} \tag{53}$$

where E_t is the energy in the transmitted signal, f is frequency, T is the integration time, and * denotes the complex conjugate. The TDOA and FDOA can be calculated simultaneously, since the function peaks when $\tau = TDOA$ and $f = FDOA$ [56]. The solution is determined by locating the peak of the CAF, $\left|CAF\left(f,\tau\right)\right|$ [24, 56]. It is worth noting that when the sensors or the emitter are not in motion that the cross-ambiguity function is simply the autocorrelation function. In Figures 85 – 88, the TDOAs, of the sensor located the farthest from the central controller and the sensor located closest to the central controller are depicted. The frequency difference of arrival is zero which is expected.

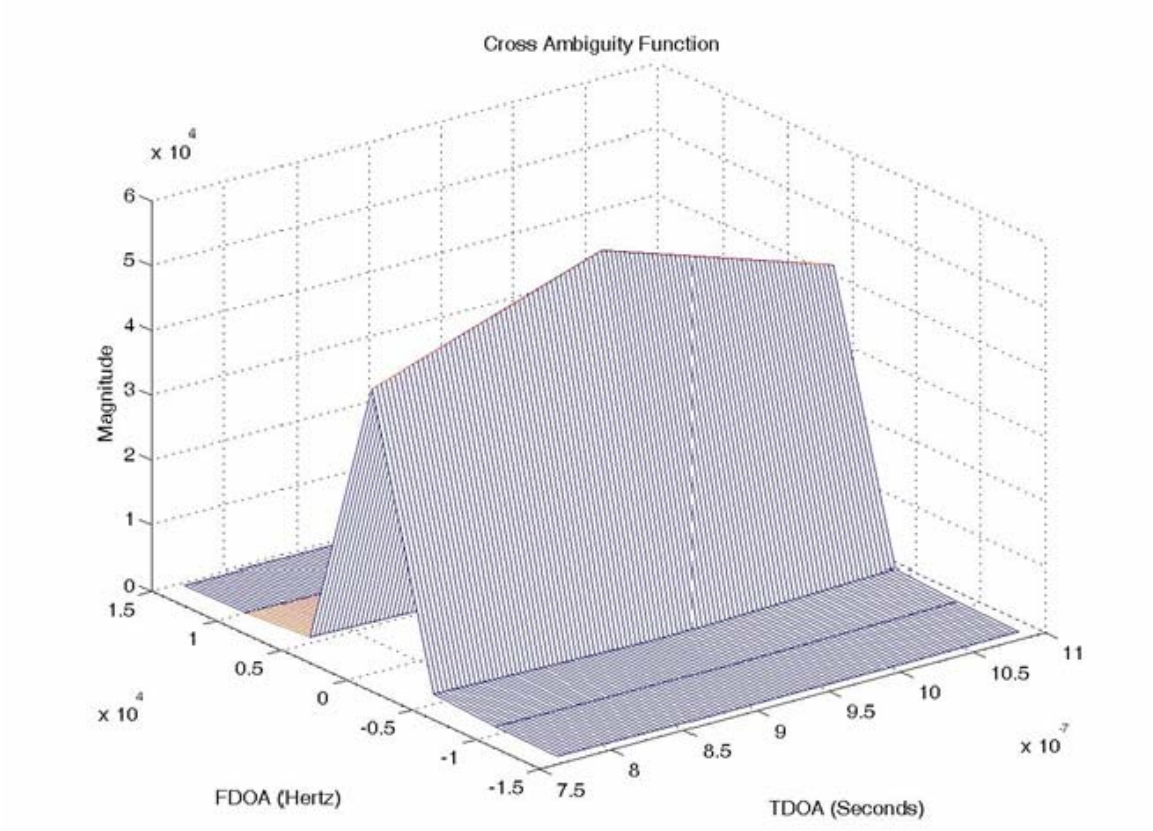


Figure 85. 3-D plot of the CAF at the maximum distance from the reference node.

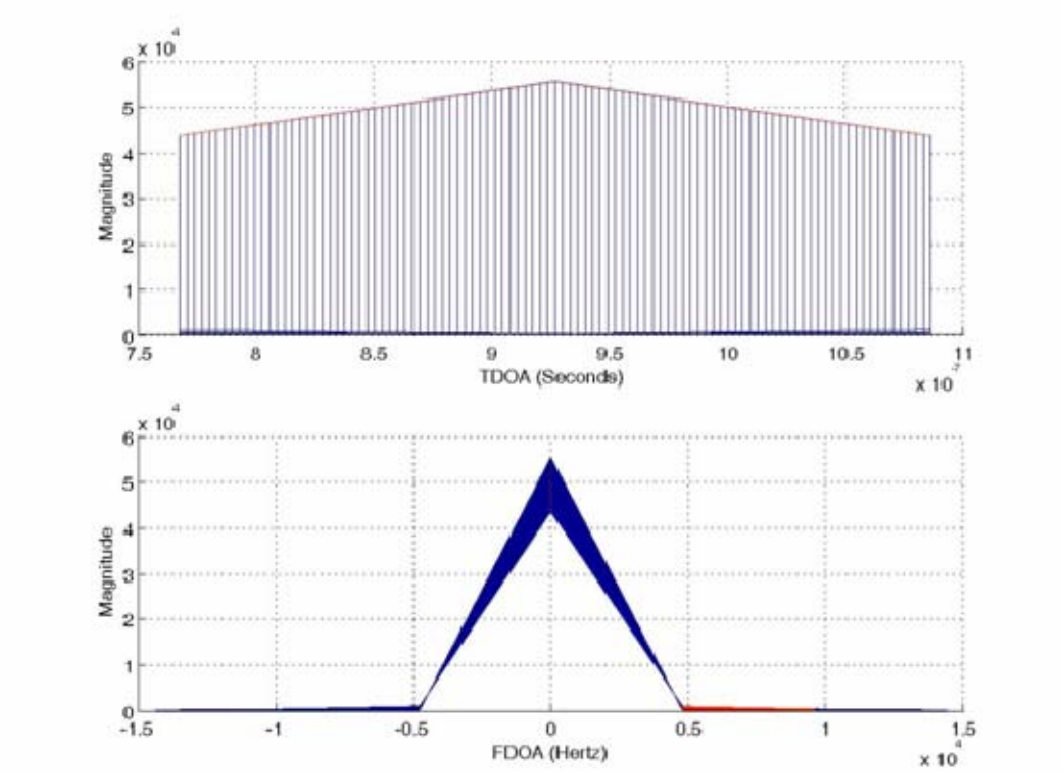


Figure 86. 2-D plot of the CAF at the maximum distance from the reference node.

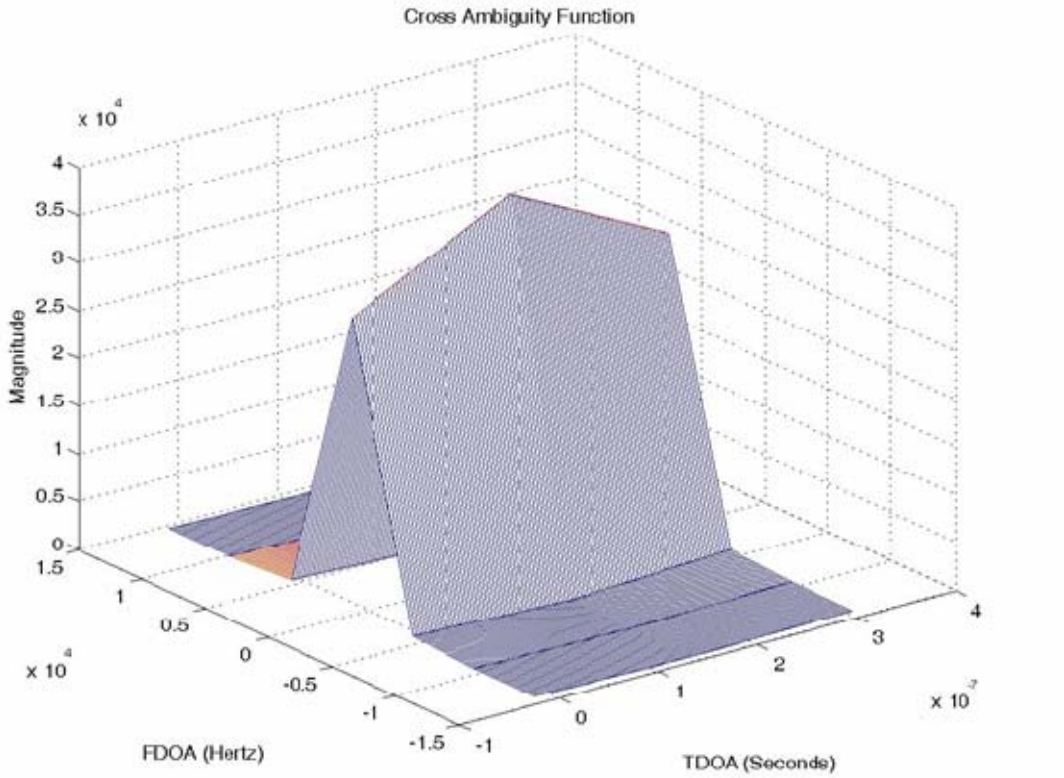


Figure 87. 3-D plot of the CAF at the minimum distance from the reference node.

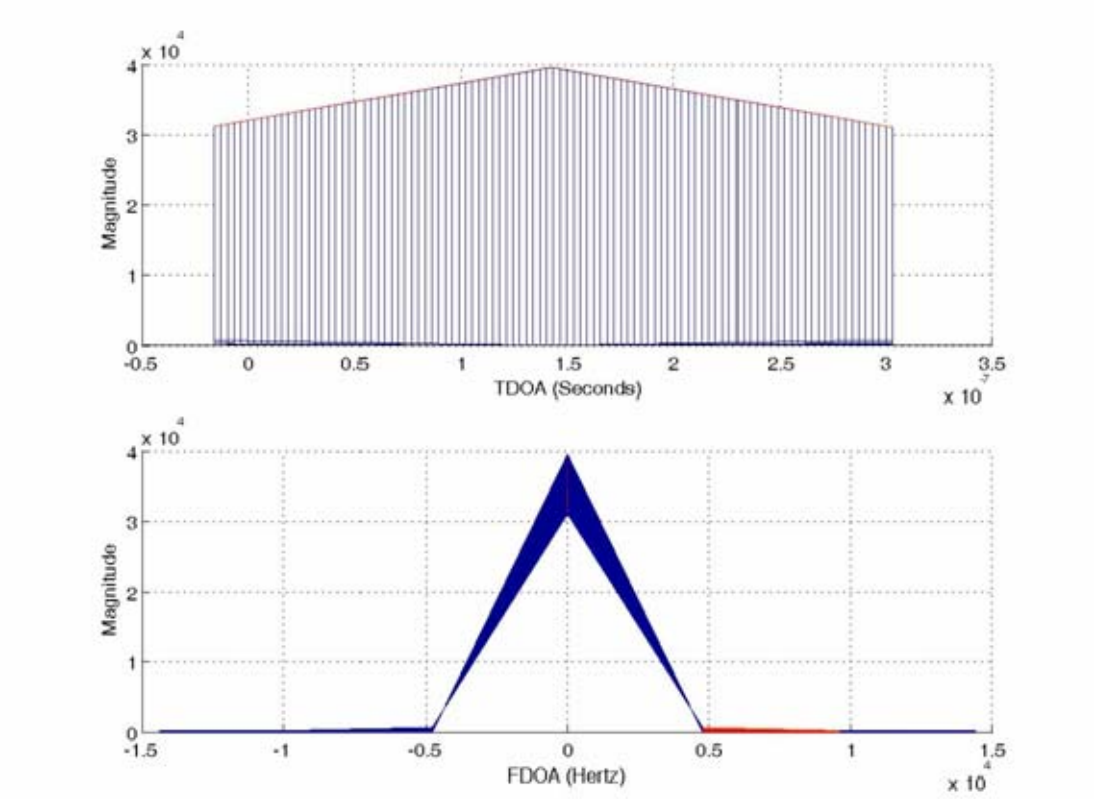


Figure 88. 2-D plot of the CAF at the minimum distance from the reference node.

The CAF is computationally expensive. Three methods were explored in [23] for computing the CAF. Due to the concern for processing capability as it pertains to power management the more efficient algorithm discussed in [23] was utilized. This algorithm was designed for use with mobile collectors and was adapted for use with the wireless SIGINT/IW antenna network. The modified MATLAB® code is in Appendices C – D [29, 32, 42].

Equation (52) can be rewritten in terms of the cross-ambiguity function as follows

$$\ln\left[p\left(\tilde{r}\,|\,\tau\right)\right] \cong \ln\left(A\right) + \frac{2A^2E_t}{N_0}\left|CAF\left(\tau,0\right)\right| \ , \tag{54}$$

where ω is zero since it is assumed that both the sensors and the target are stationary or slow moving. As defined in [72] the Cramér-Rao bound has been defined for time estimation as

$$E\left[\hat{x}(y)-x\right]^2 \geq \frac{-1}{E\left[\dfrac{\partial^2 \ln\left[p(y\,|\,x)\right]}{\partial x^2}\right]}\,, \tag{55}$$

For a more complete derivation of Equation (55) see Appendix T . Rewriting Equation (55) in terms of the CAF, the Cramér-Rao Bound becomes

$$Var\left[\hat{\tau}-\tau\right] \geq \frac{-1}{\dfrac{2A^2E_t}{N_0}E\left[\dfrac{\partial^2\left|CAF(\tau)\right|}{\partial\tau^2}\right]}\,, \tag{56}$$

In [72] the second partial derivative of the CAF is shown in Equation (57)

$$\frac{\partial^2\left|CAF(\tau)\right|}{\partial\tau^2} = \left(\frac{-1}{2E_t}\right)\int_{-\infty}^{\infty}(2\pi f)^2\left|s_t(f)\right|^2 df + \left[\left(\frac{-j}{2E_t}\right)\int_{-\infty}^{\infty}(2\pi f)\left|s_t(f)\right|^2 df\right]^2, \tag{57}$$

It can be clearly seen how the baseline distance affects the variance of the measurements. As the baseline distance increases so does the result of the CAF as depicted in figures 113 – 116. Since the CAF is in the denominator of the variance as seen in Equation (57), as the CAF gets larger the variance decreases and vice-versa.

A radian bandwidth term was defined in [72] to simplify Equation (57)

$$Var\left[\hat{\tau}-\tau\right] \geq \frac{-1}{\dfrac{2A^2E_t}{N_0}\beta^2}, \tag{58}$$

where β^2 is the radian bandwidth and is defined as follows:

$$\beta^2 \equiv \left(\frac{1}{2E_t}\right)\int_{-\infty}^{\infty}(2\pi f)^2\left|s_t(f)\right|^2 df - \left[\left(\frac{1}{2E_t}\right)\int_{-\infty}^{\infty}(2\pi f)\left|s_t(f)\right|^2 df\right]^2, \tag{59}$$

If $\left|s_t(f)\right|^2$ is assumed to be a rectangular spectrum then equation (59) can be rewritten as shown in Equation (60)

$$\beta^2 = \left(\frac{1}{2E_t}\right)\int_{-B_s}^{B_s}(2\pi f)^2\left(\frac{1}{2B_s}\right)df - \left[\left(\frac{1}{2E_t}\right)\int_{-B_s}^{B_s}(2\pi f)\left(\frac{1}{2B_s}\right)df\right]^2, \tag{60}$$

where the spectrum is defined $-1/B_s \le f \le 1/B_s$. After the integration reduces β^2 to

$$\beta^2 = \frac{2\pi^2 B_s^{\,2}}{3E_t}, \tag{61}$$

After substitution the Cramér-Rao bound becomes

$$\sigma^2{}_{TDOA} \approx \frac{1}{\frac{2A^2E_t}{N_0}\left(\frac{2\pi^2 B_s^{\,2}}{3E_t}\right)}, \tag{62}$$

Taking the square root and then canceling like terms

$$\sigma_{TDOA} \approx \frac{\sqrt{3}}{\pi B_s}\left(\frac{1}{2\sqrt{\frac{A^2W}{N}}}\right), \quad \text{where } N_0 = \frac{N}{W}, \tag{63}$$

N is defined as the noise power and W is the bandwidth of the receiver. The denominator will be further refined as follows:

$$\sigma_{TDOA} \approx \frac{\sqrt{3}}{\pi B_s\sqrt{\eta}}, \quad \text{where } \eta = \frac{4A^2W}{N}, \tag{64}$$

where η is a processing gain signal-to-noise ratio taken at the output of the receiver that is based on the signal-to-noise ratio of the received signal, integration period and the bandwidth of the received signal. B_s is the bandwidth of the signal-of-interest. Therefore the Cramér-Rao bound can be approximated

$$\sigma^2{}_{TDOA} \approx \frac{0.304}{B_s\eta}. \tag{65}$$

For multiple estimates the variance becomes

$$\sigma^2_{TDOA_\Upsilon} \approx \frac{1}{\Upsilon}\left(\sigma^2_{TDOA}\right) \approx \frac{1}{\Upsilon}\left(\frac{0.304}{B_s\eta}\right), \quad (66)$$

where Υ is the number of estimates taken. For our purposes this will be the number of sensors that participate in the solution. The importance of this variance will be discussed in more detail in the next section.

2. Newton Raphson Technique

Once the TDOAs were determined from the two-node collector pairs formed by the central controller and the sensor nodes the Newton-Raphson technique was used to calculate a geolocation position. The Newton-Raphson algorithm is based on estimation theory and uses an over-determined set of linear equations of the form

$$\underline{m} = f(\underline{z}, t), \quad (67)$$

where $\underline{m}$ is a k length vector representing the TDOA measurements and z is the location of the emitter. In the case of the SIGINT/IW sensor network $\underline{m}$ will contain the TDOAs calculated from each of the two-node collector pairs and the central controller. The size of the vector will be equal to the number of sensors, Υ, used to form the two-node collector pairs. The number of measurements is often much greater than the number of position terms.

It is envisioned that the time difference of arrival will be calculated using the cross-ambiguity function discussed earlier. The TDOA measurements in the m vector are compared with the TDOA estimates in the m_i vector, where the m_i vector is the TDOA measurements for the current estimated position as shown in Equation (67).

$$\delta\underline{m} = m - m_i = A\left(\underline{z}_{i+1} - \underline{z}_i\right) = A\delta\underline{z}_{i+1} \quad (68)$$

By iterating this process the current solution is used as the next estimated emitter location until the solution falls below some predetermined small threshold.

$$\delta\underline{z}_i = \underline{z}_{i+1} - z_i = \left[A'WA\right]^{-1}\left[A'W\right]\delta\underline{m}, \quad (69)$$

A least squares estimation is applied to yield an emitter position estimate.

3. Timing Variance

Now a discussion on the uncertainty in the TDOA estimates will be offered. Uncertainty in each TDOA measurement is caused by measurement noise and translates into uncertainty in isochrone position creating an isochrone width. The isochrone width grows with distance from the central controller. The appropriate branch depends upon the sign of the TDOA [29, 32, 42, 67].

The uncertainty is expressed as a measurement variance in Equation (65). The variance, σ^2_{TDOA} may increase due to ionospheric, atmospheric, or other effects. As will be shown later in this chapter the TDOAs are very sensitive to increases in this variation.

As can be seen in Figure 89, a sufficient receiver bandwidth is required in order to detect signals at a realistic SNR. It will be shown that the red curve (top), which corresponds to a measurement standard deviation of 1 ns is as low as the measurement standard deviation should be in order to maintain the accuracy depicted above.

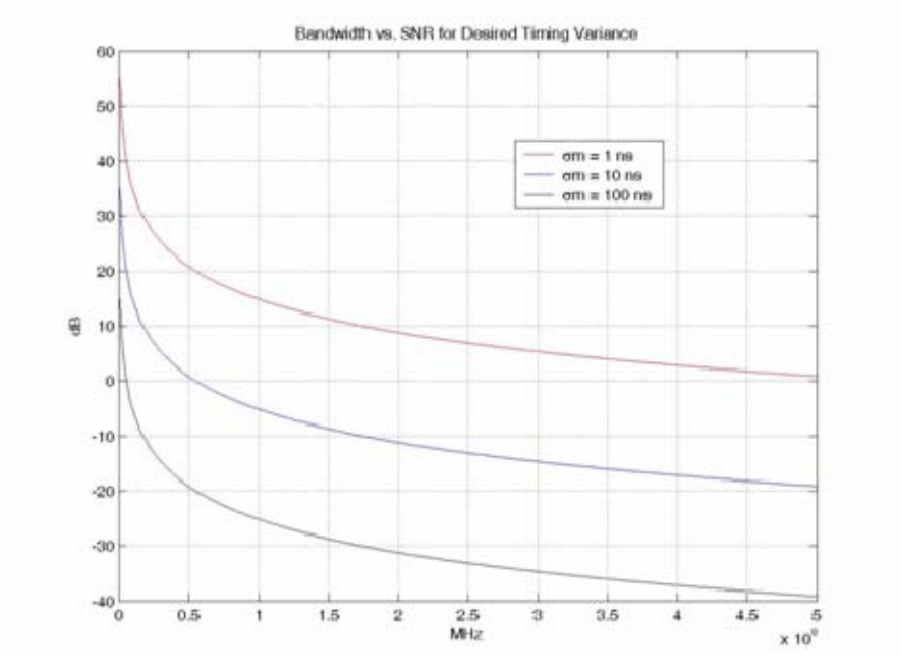


Figure 89. Receiver Bandwidth vs. SNR (dB) for a desired timing variance.

When referring to Equation (65) and Figure 89, it can be seen that the only parameters that a designer can control is the receiver bandwidth and the noise figure of the receiver, which minimizes the amount of noise the receiver adds to the received

signal. The signal power of the received signal is a function of many variables, i.e., distance from the source, multipath channels, atmospherics, etc. In all cases these are outside the control of the designer. Therefore the receiver bandwidth and noise figure of the receiver have to be chosen in order to provide a measurement standard deviation at or below 1 ns.

As a point of reference, signal-to-noise ratios have been calculated utilizing the Cramér-Rao bound Equation (65) for some COTS receivers, as seen in Table 15.

Model	**Frequency Range (MHz)**	**σ_{TDOAv}(s)**	**Receiver 3 dB Bandwidth (MHz)**	**Signal-to-Noise Ratio (dB)**
AD 8347 Direct Conversion Quadrature Demodulator	800 – 2700	1e-9	90	15.7
AD 8348 Quadrature Demodulator	50 – 1000	1e-9	500	0.8278
DRT 4011 Wideband Tuner	10 – 3000	1e-9	30	25.3
TeamSentinel Signal Acquisition Frontend (V/UHF)	30 – 300	1e-9	40	22.8

Table 15. Required received signal-to-noise rations (SNR) for a few commercially available receivers.

The AD 8347 Direct Conversion Quadrature Demodulator provides coverage within the frequency range utilized by more modern communications. The AD 8348 Quadrature Demodulator would give a better coverage range for maritime and classic military communications, i.e., push-to-talk, etc.

The measurement noise covariance is also affected by the number of TDOA measurements utilized, which can be seen in Equation (66). The accuracy can be increased by the number of TDOAs, estimates per two-node collector pair, Υ that are incorporated into the line of bearing estimate. As stated in [28] the accuracy of the line of bearing for a two-dimensional case estimate is as follows:

$$\underbrace{\delta \underline{m}}_{kx1} = \underbrace{\underline{A}}_{kx2} \underbrace{\delta \underline{z}}_{2x1}, \tag{70}$$

where $\delta \underline{m}$ is the residual vector of the TDOA estimates as defined earlier in Equation (68). In general, A requires the use of a conditioning matrix, W, because it is not a square matrix in general.

$$\underbrace{A^t W \delta \underline{m}}_{2x1} = \underbrace{A^t W A}_{2x2} \underbrace{\delta \underline{z}}_{2x1}, \tag{71}$$

W is the diagonal matrix of the inverted measurement standard deviations:

$$W = \left[\Sigma_r\right]^{-1} = \begin{bmatrix} {\sigma_0}^2 & 0 & \cdots & 0 \\ 0 & {\sigma_1}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & {\sigma_{\Upsilon-1}}^2 \end{bmatrix}^{-1}, \tag{72}$$

As discussed earlier the estimate of emitter position is solved iteratively. For clarity the z vector will be expanded to its respective *x-y* components. When noisy measurements are observed then

$$\delta \underline{m}_j = c_{j1}\delta x + c_{j2}\delta y + \iota_j; \qquad \iota_j \in N\left(0, \sigma_j\right), \tag{73}$$

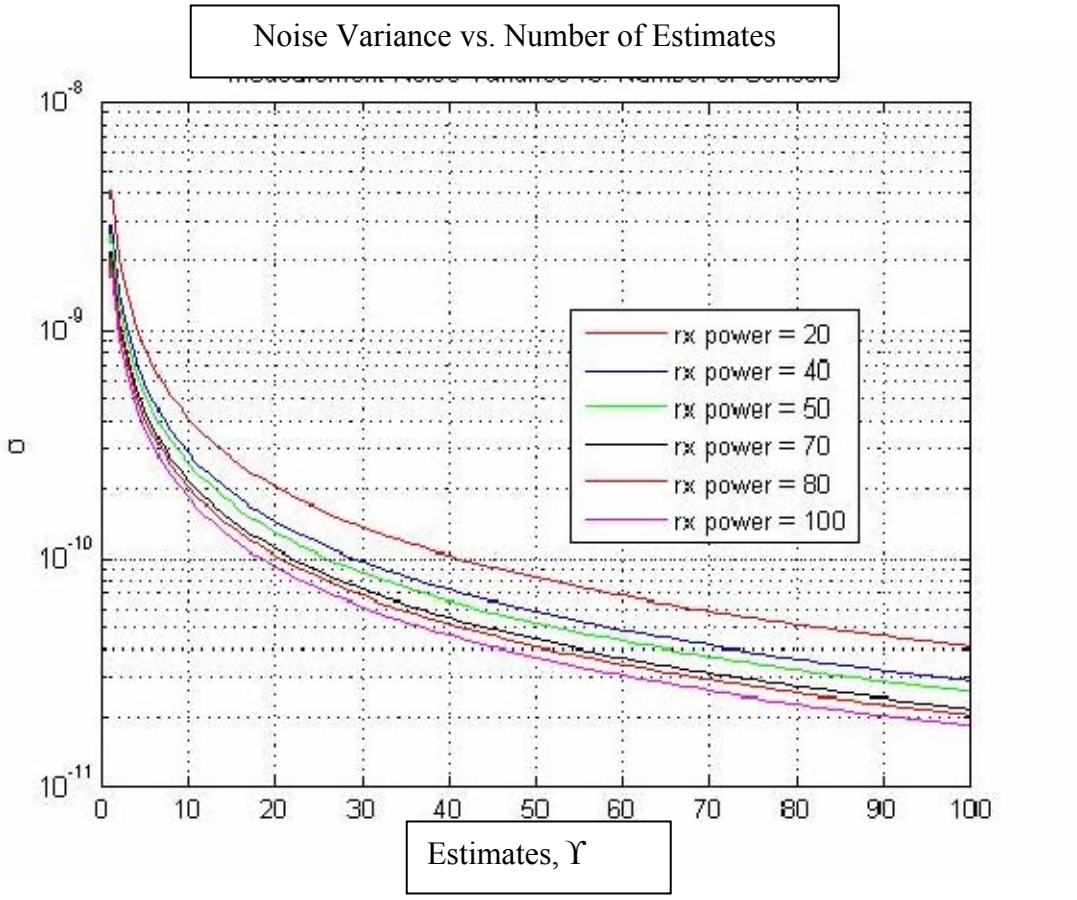
where ι_j is the measurement noise. The covariance becomes smaller as $1/\Upsilon$. As the covariance becomes smaller the less effect it has on the accuracy of the line of bearing estimate. Likewise as the number of estimates per two-node collector pair, are increased the less impact they have on reducing the covariance. The question then becomes how many TDOA, two-node collector pairs, estimates are needed and which sensors should be selected, because if we only need so many TDOA estimates the central controller needs to select the best ones. There are several methods of selecting the TDOA estimates or which sensors should be utilized. Selection methodologies is a significant research area in determining geolocation and more information can be found in [27] and [32]. For the purposes of this research the 15 estimates per two-node collector pair will be used.

The previous discussion is important to understand because the least number of estimates required in formulating a good solution means the less transmissions required across the network and less processing for the sensors and central controller, which directly translates into power and bandwidth savings. It can be surmised that the more data available to the algorithm, the more accurate the estimate as shown in the above

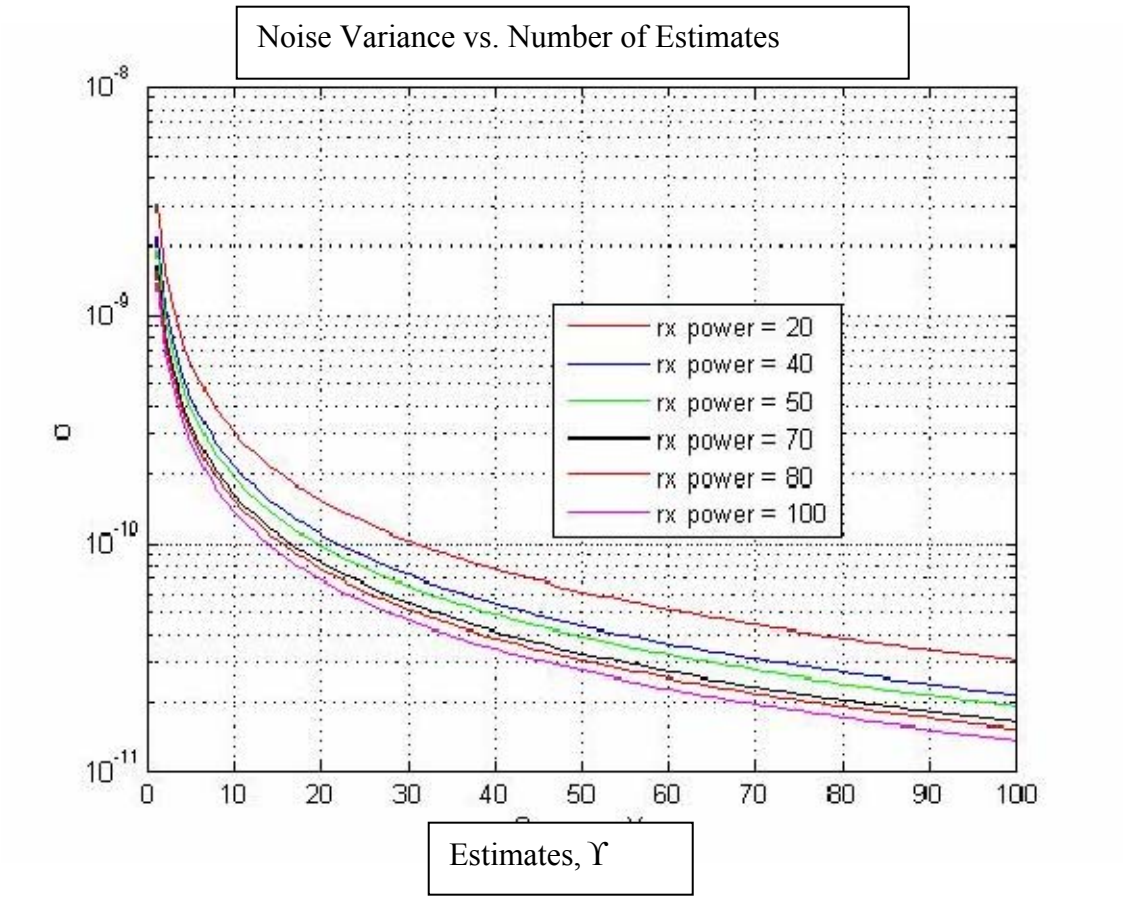
figures. The fewer the number of TDOA estimates, the less accurate the line of bearing estimate. The higher the number of TDOA estimates, the more accurate the line of bearing estimate. The tradeoff is that calculating the TDOAs is a computationally expensive process, so the greater the accuracy required the more TDOAs that have to be calculated. This relationship can be shown that for a given timing variance, σ^2_{TDOA}, only Υ estimates are required to generate the TDOA estimates. Beyond the Υ TDOA estimates the decrease in σ^2_{TDOA} diminishes to the point where no real advantage is gained. The relationship is as follows:

$$\sigma^2_{TDOA} \propto \frac{1}{\Upsilon}, \tag{74}$$

where Υ is the number of TDOA estimates, σ^2_{TDOA} is the timing variance. Three or more TDOA estimates are all that is required for the Newton-Raphson Algorithm to compute a solution. Since the accuracy is inversely proportional to the covariance and the covariance goes to zero as $1/\Upsilon$, then the more estimates that are used, the more accurate the solution. However, the amount of increase for each additional TDOA estimate would decrease and the noise measurement variance would become smaller and smaller as the number of estimates went up. This can be seen clearly in Figures 90 – 91 below.

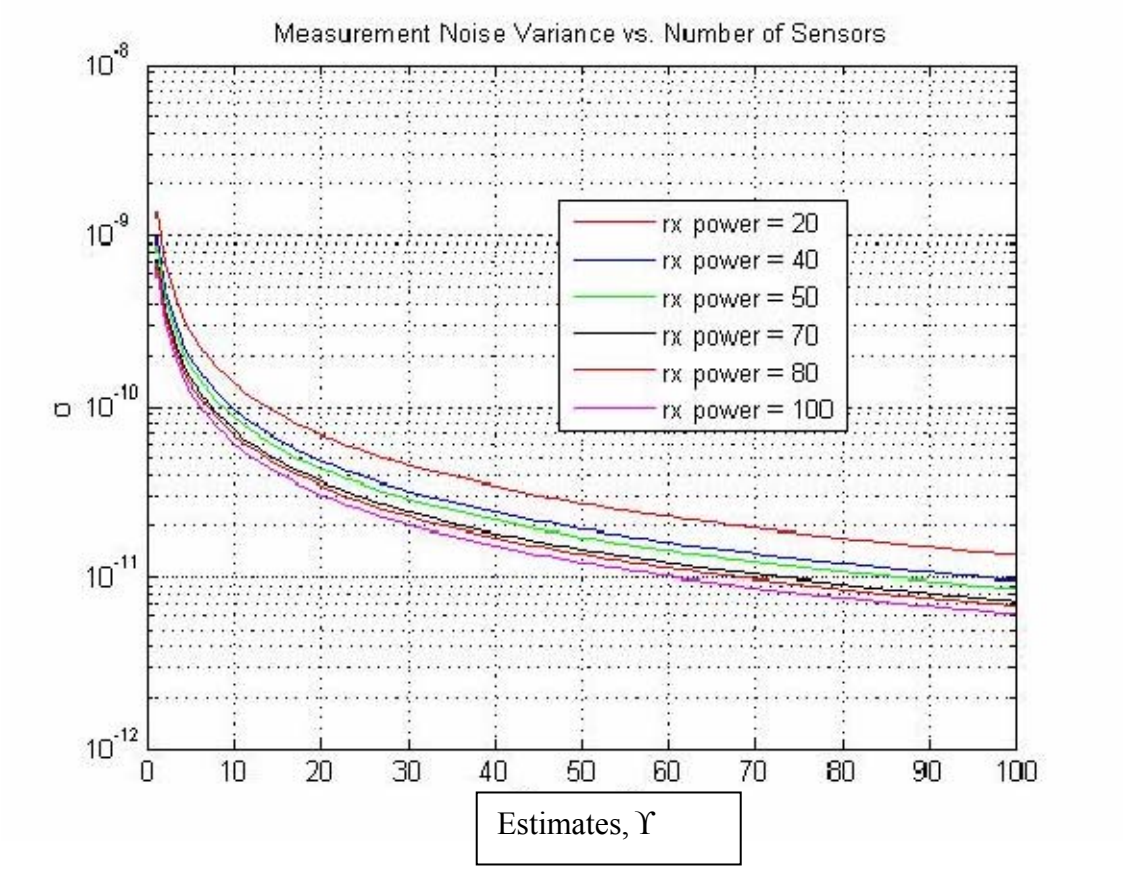

Noise Variance vs. Number of Estimates
10^-8
10^-9
10^-10
10^-11
σ
0
10
20
30
40
50
60
70
80
90
100
Estimates, ϒ
rx power = 20
rx power = 40
rx power = 50
rx power = 70
rx power = 80
rx power = 100

(a)

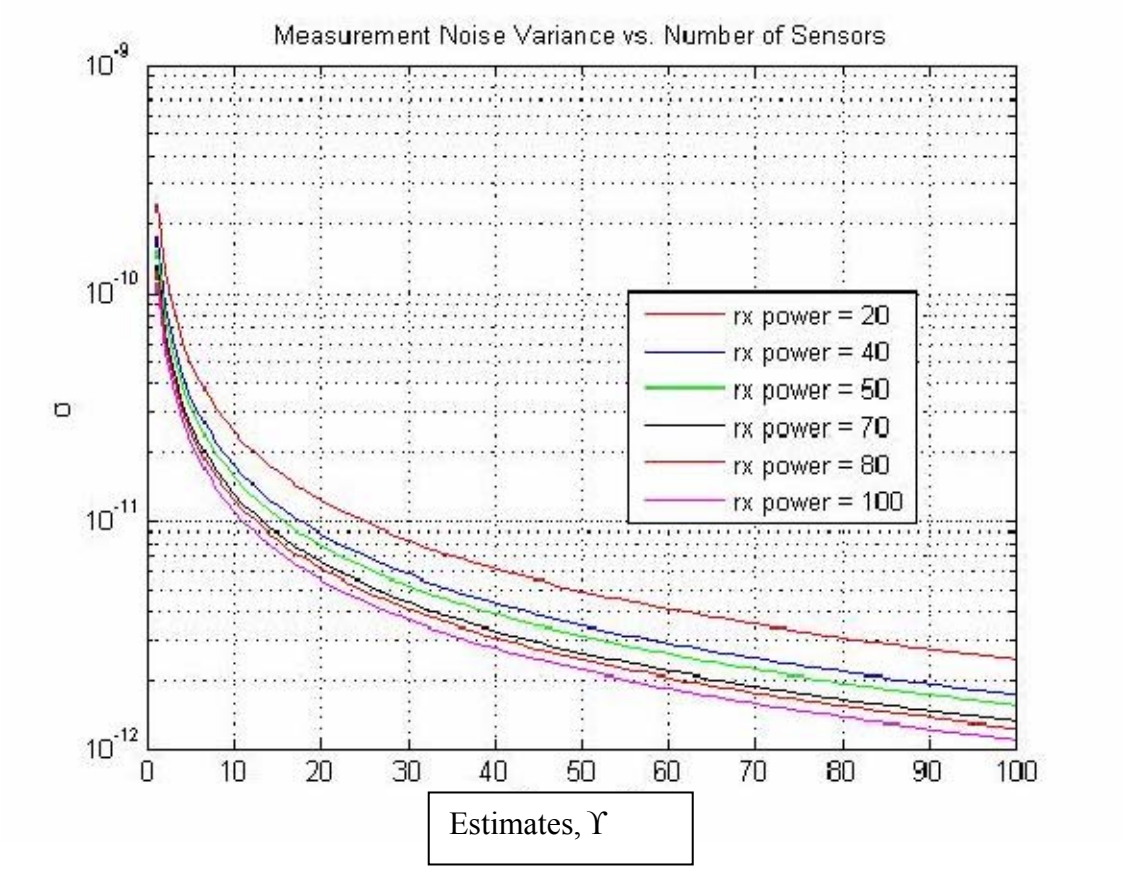


(b)

Figure 90. Measurement Noise Variance. (a) Receiver Bandwidth 30 MHz. (b) Receiver Bandwidth 40 MHz.


Measurement Noise Variance vs. Number of Sensors
10^-8
10^-9
10^-10
10^-11
10^-12
σ
0
10
20
30
40
50
60
70
80
90
100
rx power = 20
rx power = 40
rx power = 50
rx power = 70
rx power = 80
rx power = 100
Estimates, ϒ

(a)



(b)

Figure 91. Measurement Noise Variance. (a) Receiver Bandwidth 90 MHz. (b) Receiver Bandwidth 500 MHz.

Power Ratio	dB
20	13.01
40	16.02
50	16.99
70	18.5
80	19.03
100	20

Table 16. Received signal-to-noise power ratio conversion to decibels.

It can be seen from figures 90 – 91 how the received SNR lowers the number of estimates required for a given noise measurement variance. Table 16 provides a convenient conversion between power ratio and decibels. Likewise it can be seen from the figures how the received bandwidth also affects the number of estimates required. It

is interesting to note that the higher the receiver bandwidth then the number of nodes, TDOA estimates, has less of an impact. However, in all cases it can be seen that above 10 – 15 estimates the decrease in noise measurement variance begins to wane. The accuracy benefit achieved for an increase above 10 – 15 estimates is too computationally burdensome for the amount of decrease in the measurement variance. As previously mentioned the TDOA estimates are computationally expensive to calculate and minimizing this cost will assist in speeding up the solution of a line of bearing as well as reducing the processing load on the central controller. For example, using the `tic-toc` command in MATLAB® it was determined that it required 2 secs/two-node collector pair to calculate TDOA using the cross-ambiguity function. Using only 15 estimates requires 30 seconds whereas using 80 estimates would take up to 160 seconds, therefore using only 30 estimates, a 81% faster solution is provided at a noise variance loss of $3(10)^{-10}$ (at 30 MHz bandwidth). Even fewer sensors could be used but as previously stated the accuracy of the line of bearing suffers.

The fact that the grid is composed of several spatially spread two-node collectors will be leveraged to resolve the ambiguous bearing and calculate a line-of-bearing to begin beamforming. As stated earlier the CAF is computationally expensive. Even though the computations would take place at the central controller the individual nodes would need to transmit the requisite data to the central controller.

4. Line-of-bearing Estimation

The selection using this method is described as follows. The central controller will randomly select Ω nodes that are located the furthest from the central controller. The central controller will then form a two-node collector pair with each of the selected nodes.

The nodes will begin transmitting the received data back to the central controller. The central controller using the CAF, will calculate the time difference of arrival. For this method the nodes will send multiple received data samples back to the central controller. The central controller will then calculate multiple TDOA estimates for each two-node collector pair.

The total transmissions are defined as

$$T_{TDOA} = \Omega\Upsilon, \tag{75}$$

where Ω is the total number of two-node collectors and Υ is the total number of received data samples sent back to the central controller per two-node collector pair. These received data samples are then used by the Central Controller to determine the line of bearing.

If the cost per transmission is δ *joules* / *transmission*, χ_1 *joules* is the energy expended when the receiver is on, but waiting to transmit and assumes the worst case. χ_2 *joules* is the energy required to fully energize a transmitter that has been in a sleep mode and also assumes the worst case. The total energy cost to calculate a line of bearing is then defined as

$$\begin{aligned} \Gamma_{TDOA} &= \delta T_{TDOA} + (\Omega - 1)\Upsilon\chi_1 + \Omega\chi_2 \\ &= \delta\Omega\Upsilon + (\Omega - 1)\Upsilon(0.1\delta) + \Omega(0.5\delta) \end{aligned}, \tag{76}$$

where Γ_{TDOA} is in units of *joules*. The first term in Equation (76) is the energy expend the data. The second term in Equation (76), is the total energy expended waiting to transmit. This term assumes that a node awaits for all other nodes to transmit before it can transmit. χ_1 is defined in terms of a fraction of the energy expended to transmit the data. The third term is the energy expended to bring all of the participating nodes out of a sleep mode, which assumes that all participating nodes were in a sleep mode. χ_2 is also defined as a fraction of the energy to transmit the data. The energy cost per sensor is

$$\Xi_{TDOA} = \delta\Upsilon + \Upsilon(0.1\delta) - \frac{\Upsilon(0.1\delta)}{\Omega} + (0.5\delta), \tag{77}$$

where Ξ_{TDOA} is measured in units of *joules/sensor*.

A MATLAB® algorithm developed by Loomis, *et al.* at the Naval Postgraduate School was utilized. The MATLAB® routine was designed for use with high-speed mobile collectors and had to be modified for use with the wireless SIGINT/IW antenna

network. The algorithm is designed to geolocate emitters based on the TDOA measurements. The modified code can be found in Appendices F – G.

Now a specific example will be shown. The data for this example is shown in Table 17. The target is at a 45° bearing. As can be seen in Figures 92 – 94 the accuracy of the line-of-bearing estimate increases as the number of estimates from each sensor increases, which can be seen by the convergence of the line-of-bearings towards the target. This is consistent with the results from subsection 3 above that as the number of estimates increases the variance will decrease by a factor of $1/\Upsilon$.

Freq (MHZ)	**# of Trans., T_{TDOA}**	**Area, A_2 (m^2)**	**Total Energy, Γ_{TDOA} (joules)**	**# of TDOA estimates, Υ_1**	**Energy per sensor, Ξ (joules/sensor)**
157	40	2500	45δ	10	11.25δ
157	60	2500	66.5δ	15	16.625δ

Table 17. Multiple TDOA Estimates.

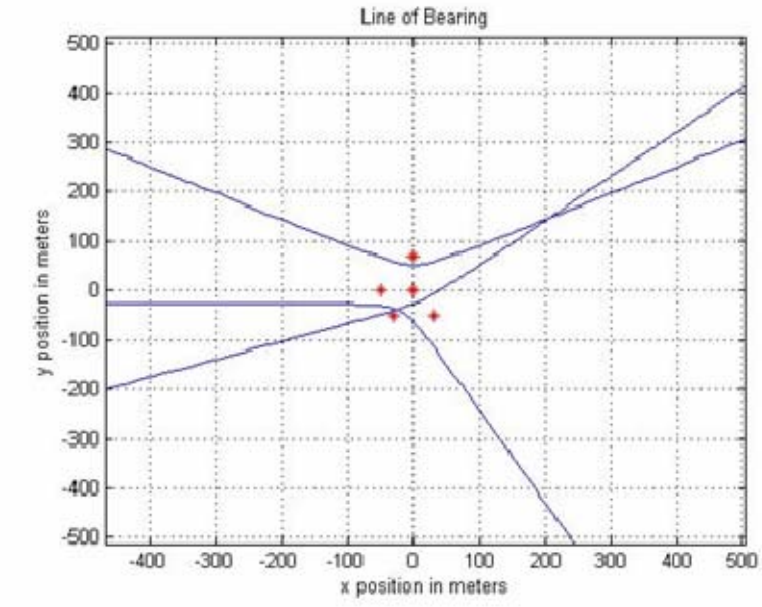


Figure 92. Isochrones from 4 two-node collectors with 5 TDOAs/two-node collector pair against an emitter at 45º

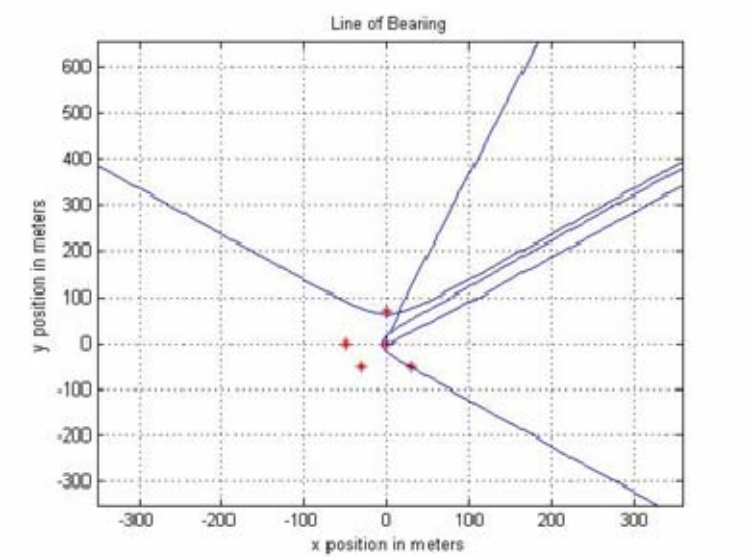


Figure 93. Isochrones from 4 two-node collectors with 10 TDOAs/two-node collector pair against an emitter at 45º

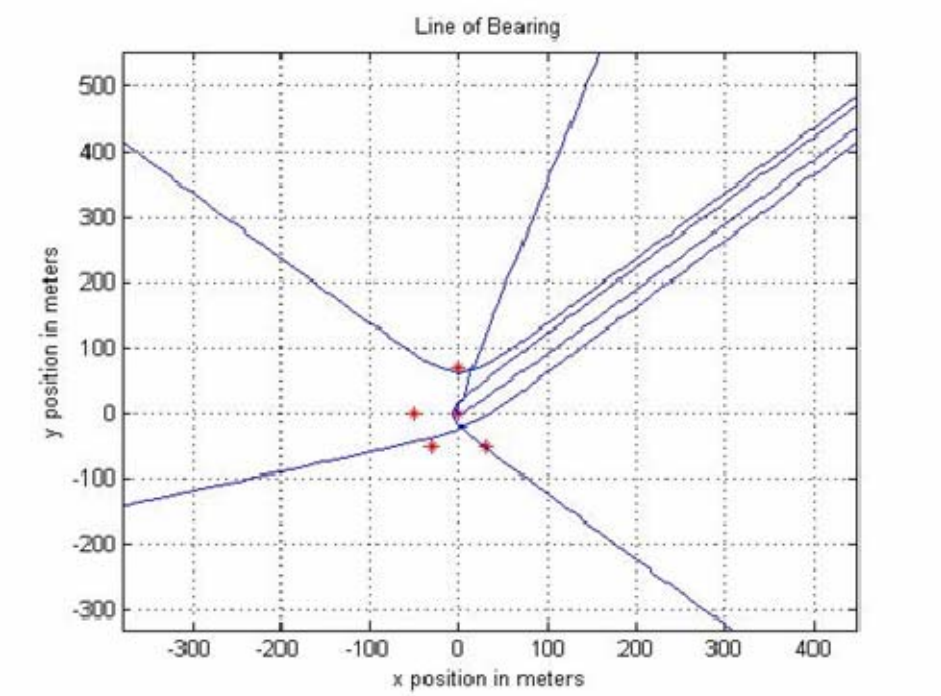


Figure 94. Isochrones from a 4 two-node collectors with 15 TDOAs/two-node collector pair against an emitter at 45º

It can be observed in the above figures that the isochrones from the two-node collector pairs begin to converge towards the target. As expected as the number of estimates increases the lines-of-bearing get more accurate.

This method would obviously be able to provide a line of bearing to the target to facilitate follow-on beamforming for enhanced collection. Consequently, a proposed approach for a wireless antenna sensor grid using TDOA to determine the line of bearing to the target is as follows:

For this method the individual steps are listed below:

Step 1. The central controller acts as a reference antenna by gaining initial intercept and frequency-of-interest (FOI) determination.

Step 2. The central controller will then coordinate the direction-of-arrival (DOA) determination using n number of sensors. The central controller will calculate Υ TDOA estimates per Ω two-node collector pairs. The central controller will utilize the cross-ambiguity function presented earlier to calculate the TDOAs between each of these Ω sensors and the central controller.

Step 3. The central controller will then use the modified Newton-Raphson technique to determine a line-of-bearing to the target.

Step 4. The central controller will then begin the beamforming process. Section B will cover in detail two various methods for adaptive beamforming.

The TDOA determination process would take a total of t_{TDOA} seconds.

$$t_{TDOA} = \Omega\varsigma\Upsilon + \Omega\varepsilon + \Omega\Upsilon t_1 + \Omega t_2, \tag{78}$$

where Ω is the number of nodes, ς is the cost per TDOA estimate to perform the CAF, Υ is the number of TDOA estimates per two-node collector pair and ε is the time required to calculate the line of bearing using the NRT. The time a node takes to transmit is t_1. This term assumes a TDMA type medium access, in which a node has a time slot and awaits for all other nodes to transmit before it can transmit. The time required to awake the nodes is t_2 and assumes the worse case, that all nodes selected are in a sleep mode.

A method for computing TDOAs and then subsequent lines-of-bearing has been presented. As was shown the CAF and Newton-Raphson method as implemented was able to determine the line of bearing to the target. Now the central controller would be

able to adaptively form beams in the direction of the target to enhance collection and/or offensive information operations. In the next section, beamforming will be considered as an alternative for discussion.

B. ADAPTIVE BEAMFORMING

The primary intent of using beamforming is to enhance collection and/or offensive information operations by taking advantage of the number of elements and having them work coherently to increase their respective detection and transmission ranges. However, as mentioned earlier the azimuth of the target needs to be known a priori in order for the central controller to know where to point the beam. In Section A, a proposal was offered for using TDOA to determine the target azimuth. Now a discussion will be presented on using beamforming as a method of determining the azimuth of the target by scanning the beam analogous to a radar searching for the azimuth with the highest received power.

The beamformer method is the most basic method employed in direction of arrival estimation with an array antenna. A central controller can scan the beam electrically in a 360° arc, wherein the main beam scans and determines the direction in which the output power of the array becomes a maximum [18, 69]. The directions that give large power outputs can then be taken as the estimated direction of the target signals. When the power is plotted against these directions, it exhibits a peak in the direction of a target signal. Depending on the type of beamformer used, many different methods can be used [69, 71].

Almost all DOA estimation algorithms, whether narrowband or wideband, use narrowband beamforming techniques to estimate the DOA for the many different frequency bands. These separate estimates are then combined to get one estimate based on feasible statistical observations. [71] An example to illustrate this point will build on the discussions from Chapter IV, Section B, subsection 3. The narrowband beamformer from Equation (22) is rewritten for convenience and becomes Equation (79).

$$b(t) = \underline{w}^H \underline{s}(t), \tag{79}$$

where the weighting vector $\underline{w}$ emphasizes one particular direction. Given samples $b(1), b(2), \ldots, b(N)$, the output covariance is

$$P(w) = \frac{1}{N}\sum_{t=1}^{N}\left|b(t)\right|^2 = \frac{1}{N}\sum_{t=1}^{N}\underline{w}^H \underline{s}(t)\,\underline{s}^H(t)\,\underline{w} = \underline{w}^H R_{SS}\,\underline{w} \tag{80}$$

Where R_{SS} is obtained as

$$R_{SS} = \frac{1}{N}\sum_{t=1}^{N}\underline{s}(t)\,\underline{s}^H(t) \tag{81}$$

Many different choices of the weighting vector $\underline{w}$ can be made leading to different properties of the beamforming schemes [62, 70].

As discussed in [5] and [62] the weighting vector for the conventional beamformer is chosen in order to maximize the received power in a certain direction ϕ, as

$$\underline{w}_{BF} = \frac{\underline{\Psi}(\phi)}{\sqrt{\underline{\Psi}^H(\phi)\,\underline{\Psi}(\phi)}} \tag{82}$$

the classical spatial spectrum is obtained

$$P_{BF}(\phi) = \frac{\underline{\Psi}^H(\phi)\,R_{SS}\,\underline{\Psi}(\phi)}{\underline{\Psi}^H(\phi)\,\underline{\Psi}(\phi)} \tag{83}$$

This is commonly referred to as the Bartlett beamformer [5, 58, 62].

As discussed earlier, our wireless phased array will consist of SIGINT sensor nodes that are deployed randomly over an area of interest, resulting in random positioning of the sensor nodes. Subsequently, these sensor nodes form clusters and their locations are determined using location discovery techniques. These locations are then reported back to the central controller. In addition, the SIGINT/IW antenna elements will in some cases not be located within a half-wavelength from each other. Therefore an algorithm capable of mitigating the grating lobes discussed in Section A, subsection 1 will have to be employed. The array will need to be able to scan 360° searching for target signals. Because of these factors the pattern of the array will need to be controlled dynamically. Therefore, an adaptive algorithm will be implemented. The azimuth that

gives the largest power output can then be taken as the estimated DOA of the incident signals [66, 69].

Once the target has been localized the central controller can enhance the gain toward the desired signals and can null toward undesired signals. Correct control of beams and nulls is indispensable for the adaptive antenna. The pattern of the array will be controlled by dynamically varying the phase and amplitude of the received signal of each element.

Various adaptive algorithms exist, such as the least mean squares (LMS), recursive least squares (RLS) and constant modulus algorithm (CMA). The LMS algorithm recursively computes and updates the complex weights. [27, 73] The LMS algorithm requires knowledge of the desired signal, which will be assumed. This research will build on the research conducted in [68] using the LMS algorithm. Figure 95 shows the block diagram of a narrow-band adaptive beamformer using the LMS algorithm. In addition to being able to steer the main beam towards the signal of interest, the LMS algorithm can also simultaneously suppress interfering signals through the calculation of the complex weights. The grating lobes discussed earlier are also a consideration in determining which method to employ in determining the DOA. It will be shown that the LMS algorithm overcomes the effect of these lobes by eliminating all periodicities in the element locations by randomly selecting nodes to form the beam. In [57] this process is referred to as thinning the array.

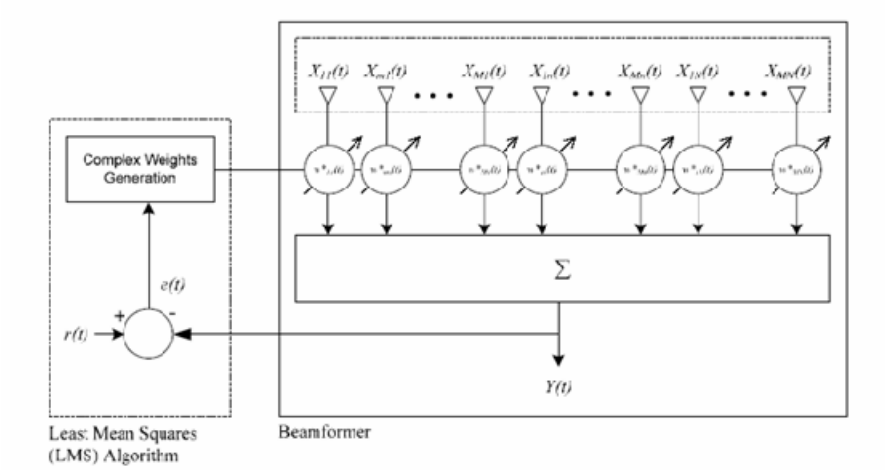


Figure 95. M×N narrow-band adaptive beamformer. [From Ref. 27]

The weights are calculated by analysis of the error between the observed and desired signal. These weights are used to adaptively form the beam in the desired direction [69, 73]. The error is defined to be:

$$e(t) = \alpha(t) - b(t), \tag{84}$$

where $\alpha(t)$ is the reference signal and $b(t)$ is the output of the beamformer. In practice, $\alpha(t)$ needs to be closely correlated to the desired signal $s(t)$.

As the signal is received by the individual array elements the complex weights are iteratively adjusted and applied. The complex weights are chosen to minimize the mean-square error between the beamformer output $b(t)$ and the reference signal $\alpha(t)$ for each iteration. This is shown in Equation (85) [27, 69, 73].

$$E\left[e^2(t)\right] = E\left[\left(\alpha(t) - b(t)\right)^2\right], \tag{85}$$

which can be expanded to the form given by

$$E\left[e^2(t)\right] = E\left[\alpha^2(t)\right] - 2\underset{\sim}{w}^H E\left[\alpha(t)\underset{\sim}{s}(t)\right] + \underset{\sim}{w}^H E\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]\underset{\sim}{w}, \tag{86}$$

where $E[\square]$ is the expectation operator. The minimum mean-square error (MMSE) is obtained by setting the gradient vector of Equation (85) with respect to $\underset{\sim}{w}$ to zero, giving

$$\nabla_{\underset{\sim}{w}} E\left[e^2(t)\right] = -2E\left[\alpha(t)\underset{\sim}{s}(t)\right] + 2E\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]\underset{\sim}{w}, \tag{87}$$

As per [27, 69, 73], the value of the weight vector at time $(t+1)$ is updated based on the method of steepest descent as follows

$$\underset{\sim}{w}(t+1) = \underset{\sim}{w}(t) + \frac{1}{2}\mu\left[-\nabla_{\underset{\sim}{w}} E\left[e^2(t)\right]\right], \tag{88}$$

where μ is the convergence factor. By substituting $\nabla_{\underset{\sim}{w}} E\left[e^2(t)\right]$ from Equation (87) into the above equation, we have

$$\begin{aligned}\underset{\sim}{w}(t+1) &= \underset{\sim}{w}(t) + \frac{1}{2}\mu\left[2E\left[\alpha(t)\underset{\sim}{s}(t)\right] - 2E\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]\underset{\sim}{w}\right] \\ &= \underset{\sim}{w}(t) + \mu\left[\alpha(t)\underset{\sim}{s}(t) - \underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\underset{\sim}{w}\right] \\ &= \underset{\sim}{w}(t) + \mu\underset{\sim}{s}(t)\left[\alpha(t) - \underset{\sim}{s}^H(t)\underset{\sim}{w}\right]\end{aligned}$$

which can be simplified to become:

$$\underset{\sim}{w}(t+1) = \underset{\sim}{w}(t) + \mu\underset{\sim}{s}(t)e^*(t), \tag{89}$$

where * represents complex conjugation and the instantaneous estimates given by $\left[\alpha(t)\underset{\sim}{s}(t)\right]$ and $\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]$ have been used instead of $E\left[\alpha(t)\underset{\sim}{s}(t)\right]$ and $E\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]$, respectively. As stated in [27, 69, 73], the term, $E\left[\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)\right]$ is referred to as the array correlation matrix R_{SS} and implies a correlation between the signals received by the various array elements exists. The convergence factor μ is related to the eigenvalues of $\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)$. If the eigenvalues of $\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)$ are widely spread, convergence will be slow. The value of μ needs to be in the range shown in Equation (91) for convergence of the LMS algorithm [19, 73]

$$0 < \mu < \frac{2}{\lambda_{\max}}, \tag{90}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\underset{\sim}{s}(t)\underset{\sim}{s}^H(t)$.

The following is an iteration of the LMS adaptive beamforming algorithm. For our sensor grid, the central controller selects the number of array elements needed to form the desired beam. The central controller in effect creates an array, which is merely a subset of the total array. For example, a SIGINT grid that contains K elements randomly spread over an area, A_1. The central controller will create a subset of Λ elements

randomly spread over an area, A_2, where $\Lambda \le K$ and $A_2 \le A_1$. The central controller will then select P nodes contained within an area, A_2, where $P \square \Lambda$.

$$\Psi(\theta,\phi) = e^{\left(j\frac{2\pi}{\lambda}*\left(x\sin(\theta)*\cos(\phi)+y\sin(\theta)\sin(\phi)\right)\right)}, \tag{91}$$

where λ is the wavelength of the signal in meters and x and y are the location of the individual antenna elements in meters. θ is the elevation angle and ϕ is the azimuth angle.

The signals collected at these nodes are forwarded back to the central controller. The received signal is shown below.

$$b(t) = \Psi(\theta,\phi) * s(t) + \underline{n}(t), \tag{92}$$

where

$$s(t) = A(t)e^{-j\left(\frac{2\pi f_0 t}{f_{samp}}+\gamma(t)\right)}, \tag{93}$$

where f is the frequency of interest, t is the sample increment, f_{samp} is the sampling rate, $\gamma(t)$ is the phase and $n(t)$ is additive white Gaussian noise. Twice the frequency of interest was used as the sampling rate and five hundred data samples were used. The reference signal $\alpha(t)$ needs to be closely correlated with $s(t)$.

The central controller will then calculate the array weights $\underline{w}$ by summing each element over all possible values of θ and ϕ.

$$\Psi(\theta,\phi) = \sum_{m=1}^{M}\sum_{n=1}^{N} w^*_{mn} e^{j\frac{2\pi}{\lambda}\left(x_{mn}\sin(\theta)\cos(\phi)+y_{mn}\sin(\theta)\sin(\phi)\right)}, \tag{94}$$

where $P = MxN$ the total number of nodes in the subset and w_{mn} is the complex weight

$$w^*_{mn} = W_{mn} e^{j\frac{2\pi}{\lambda}\left(x_{mn}\sin(\theta_a)\cos(\phi_0)+y_{mn}\sin(\theta_a)\sin(\phi_0)\right)}, \tag{95}$$

applied to the $(m,n)^{th}$ element. The maximum value of $\Psi(\theta,\phi)$ occurs at $(\theta,\phi) = (\theta_a,\phi_0)$, and the main lobe points towards (θ_a,ϕ_0).

Each data sample of the signals collected from the chosen nodes iteratively updates the corresponding array weights. The magnitude and angle of $\Psi(\theta_a, \phi_0)$ are stored. This completes an iteration of the adaptive beamforming solution.

Should the central controller repeat this process using new data collected from the same P nodes or should the central controller request information from a different P node subset? Using the same P nodes could potentially decrease the network burden and reduce the number of required antenna elements. However, randomly selecting the nodes for each subset could distribute the power management across the network, as well as, lower the computational burden of the central controller.

If the first option is chosen then the central controller simply takes another set of data samples from each of the P nodes used in the previous calculation. The process described earlier is repeated and the final result is added to the previously stored result.

If the second option is chosen then the central controller takes another set of data samples from a new subset of P nodes. Then the process described earlier is repeated and the final result is added to the previously stored result. This magnitude is then plotted in dB against all possible values of azimuth and elevation.

A third option is possible, as well. In this option the central controller will simply take another set of data samples from each of the P nodes similar to the first calculation. However, in this case the nodes will have moved in the intervening time interval. This will have the affect of creating a larger antenna field than truly exists.

It will be shown that the second method is a much better way to calculate the weights. The first method is unable to overcome the grating lobes and provide a sufficient beam for direction of arrival determination. The second method because it is selecting a different subset of nodes for each iteration is able to overcome the grating lobes and provide a sufficient beam for direction of arrival determination with a minimal number of iterations. It will be shown that having mobile sensors will give an unparalleled surveillance capability with only a few sensors. However, it is assumed that these mobile sensors will be high cost and able to recharge their batteries at will.

The selection process is put into a general equation shown in Equation (96). How the central controller takes another set of data samples is what will be shown in subsequent paragraphs. Once the central controller selects more data the process described earlier is repeated and the final result is added to the previously stored result.

$$\Psi_{Total}(\theta,\phi) = \sum_{i=1}^{L} \Psi_i(\theta,\phi), \tag{96}$$

where L equals the total number of subsets created. Substituting Equation (95) into Equation (96) results in the following:

$$\Psi_{Total}(\theta,\phi) = \sum_{i=1}^{L}\left[\sum_{m_i=1}^{M}\sum_{n_i=1}^{N} w^*_{m_i n_i} e^{j\frac{2\pi}{\lambda}\left(x_{m_i n_i}\sin(\theta)\cos(\phi)+y_{m_i n_i}\sin(\theta)\sin(\phi)\right)}\right], \tag{97}$$

where $P = MxN$ the total number of nodes in each subset. This magnitude is then plotted in dB against all possible values of azimuth and elevation. This process is repeated until a desired solution has been found.

1. Method One

The selection using this method is described as follows. The central controller will randomly select a subset of P nodes from a total of Λ elements that are contained within an area, A_2. As stated earlier, the overall array contains a total number of nodes, K, contained within and area, A_1, where $P \leq \Lambda \leq K$ and $A_2 \leq A_1$. These P nodes will form a subset of array elements that the central controller will use to calculate the array weights. The LMS algorithm adaptively produces weights for the formation of the beam. These weights are then used to calculate the magnitude and angle of the beam. Next the central controller obtains more data from the same subset of P nodes selected earlier. The new weights are then calculated and used to calculate the magnitude and angle of the beam in the desired direction. This result is then added to the previous calculations and then stored. This iterative process is continued for L_1 iterations where L_1 is used to represent Method One. Equation (97) is rewritten dropping the *i* dependency, to reflect the fact that the same P nodes are being used. This process is shown in Equation (98).

$$\Psi_{Total}(\theta,\phi) = L_1\left[\sum_{m=1}^{M}\sum_{n=1}^{N} w^*_{mn} e^{j\frac{2\pi}{\lambda}\left(x_{mn}\sin(\theta)\cos(\phi)+y_{mn}\sin(\theta)\sin(\phi)\right)}\right], \tag{98}$$

where $P = MxN$ the total number of nodes in each subset. This magnitude is then plotted in dB against all possible values of azimuth and elevation. This process is repeated until a desired solution has been found. MATLAB® code developed in [73] was modified to conduct the simulations in this section. This modified code can be found in Appendix I.

MATLAB® code in [73] was developed to randomly distribute sensors within a square grid. For ease of simulation within MATLAB® the number of elements M was equal to the number of elements N. Once this distribution of sensors was completed the central controller began to calculate the beam. As previously, stated the central controller continued to use the data received at the same P nodes. Several simulations were conducted to investigate the viability of this method. The results are as follows:

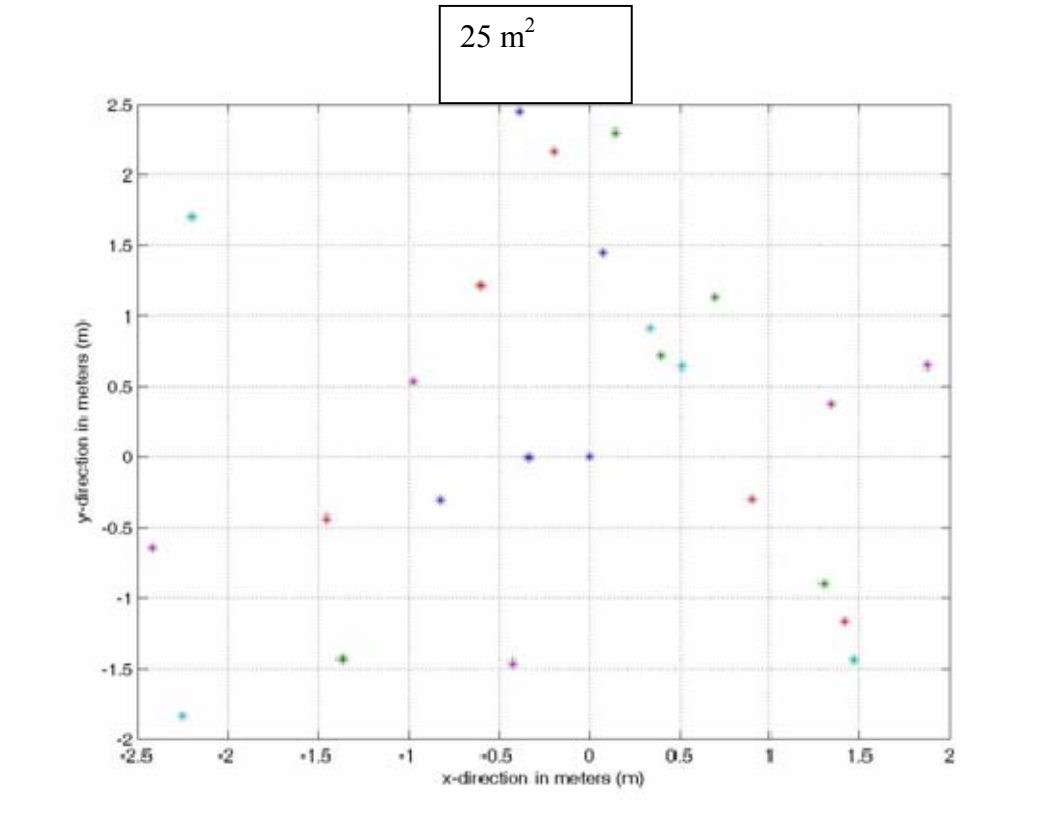


Figure 96. 25 Node array at a node density of 1 sensors/m^2.

The test parameters for this section assumed no errors and no node failures; elevation and azimuth 45°. Interferers were placed at 18° with elevation and azimuth at

45°. The number of nodes in the subset P_i was selected to be 25. The number of iterations L_1 was varied between 10, 20 and 50 as shown in Table 18. The area, A_2 was varied between 25, 400, 2500 and 250000 m^2. As was shown in Figures 32 – 34, as a given number of nodes are spread over an increasing area the average distance between sensors gets larger, therefore decreasing the node density. It can be observed in Equations (24) – (27) as the frequency goes up the beamwidth gets smaller and this was illustrated in Figures 31 (a) – (c). Therefore, by analyzing Equations (24) – (27) there are two factors that have a big impact on the beamwidth, one is the node density and the other is the frequency of the signal-of-interest. These factors will now be illustrated through several examples. The pertinent information for the simulations is shown in Tables 18 – 20. The area, A_2 of 25 m^2 is shown in Figure 96.

Freq (MHz)	**Noise Power (pW)**	**Area, A_2 (m^2)**	**Sample Size**	**# of iterations L_1**	**Number of Sensors per subset, P**	**Node Density (sensors/m^2)**
157/800/2400	1	25	500	10	25	1
157/800/2400	1	25	500	20	25	1
157/800/2400	1	25	500	50	25	1

Table 18. Adaptive beamforming utilizing Method One with a node density of 1 sensor/m^2.

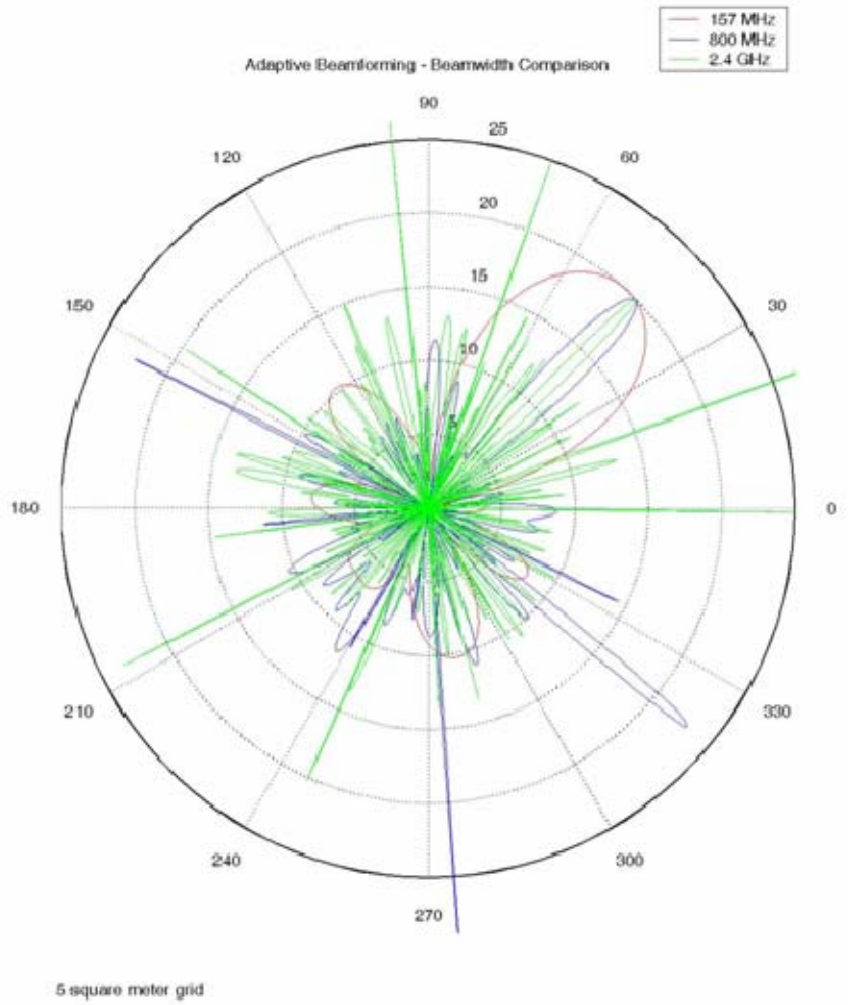


Figure 97. The Array Pattern determined utilizing Method One with L_1 = 10 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As can be seen in the Figure 97 the beamwidth gets smaller as the frequency goes up, which is consistent with the results shown in Figures 31 (a) – (c). This can also be verified by observing the relationship of wavelength and node density to the beamwidth illustrated in Equations (24) – (27). Therefore for a given node density the elements are spaced at more wavelengths apart for the high frequencies. Also present at the higher

frequencies are a lot of side lobes that would need to be resolved if this node density was used. Ten iterations does not appear to be sufficient for this node density.

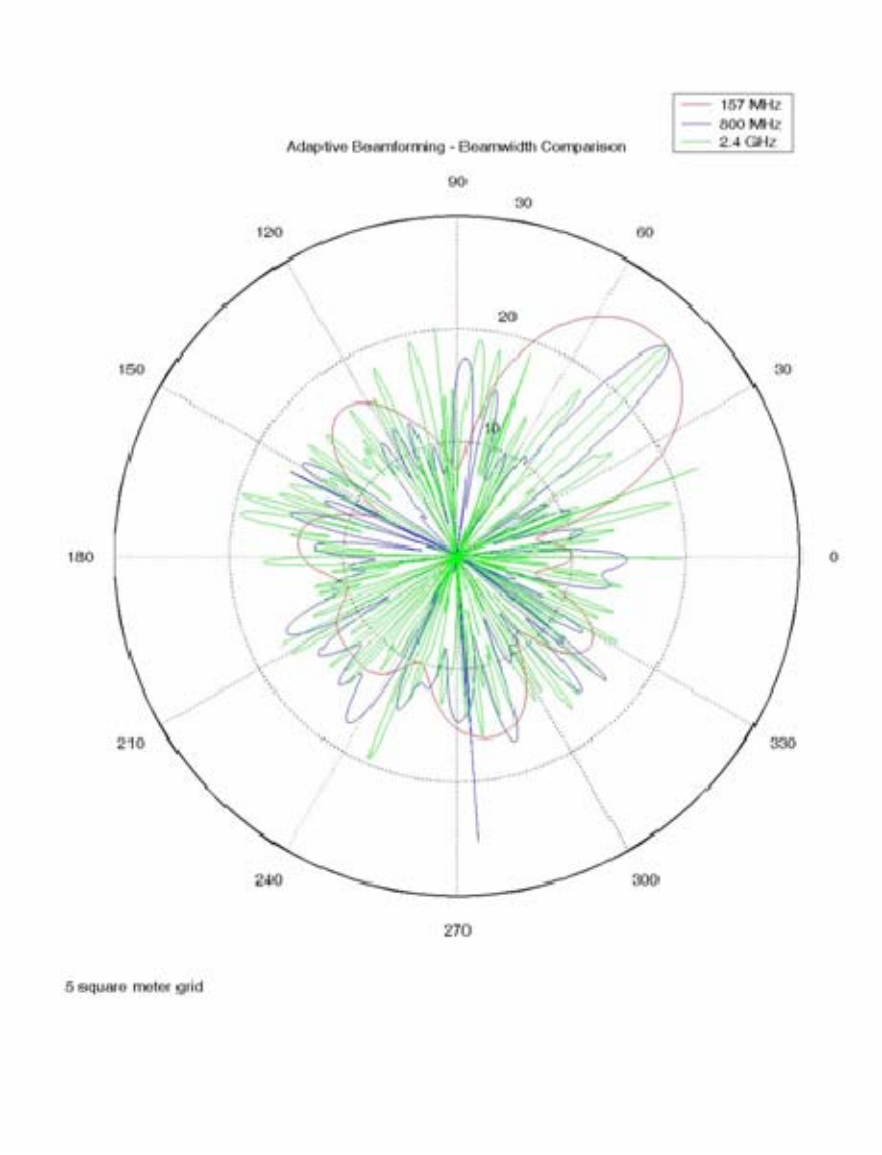


Figure 98. Array Pattern determined utilizing Method One with $L_1 = 20$ iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As can be seen from Figure 98, 20 iterations provided a somewhat cleaner solution, but still insufficient to provide direction of arrival determination capability.

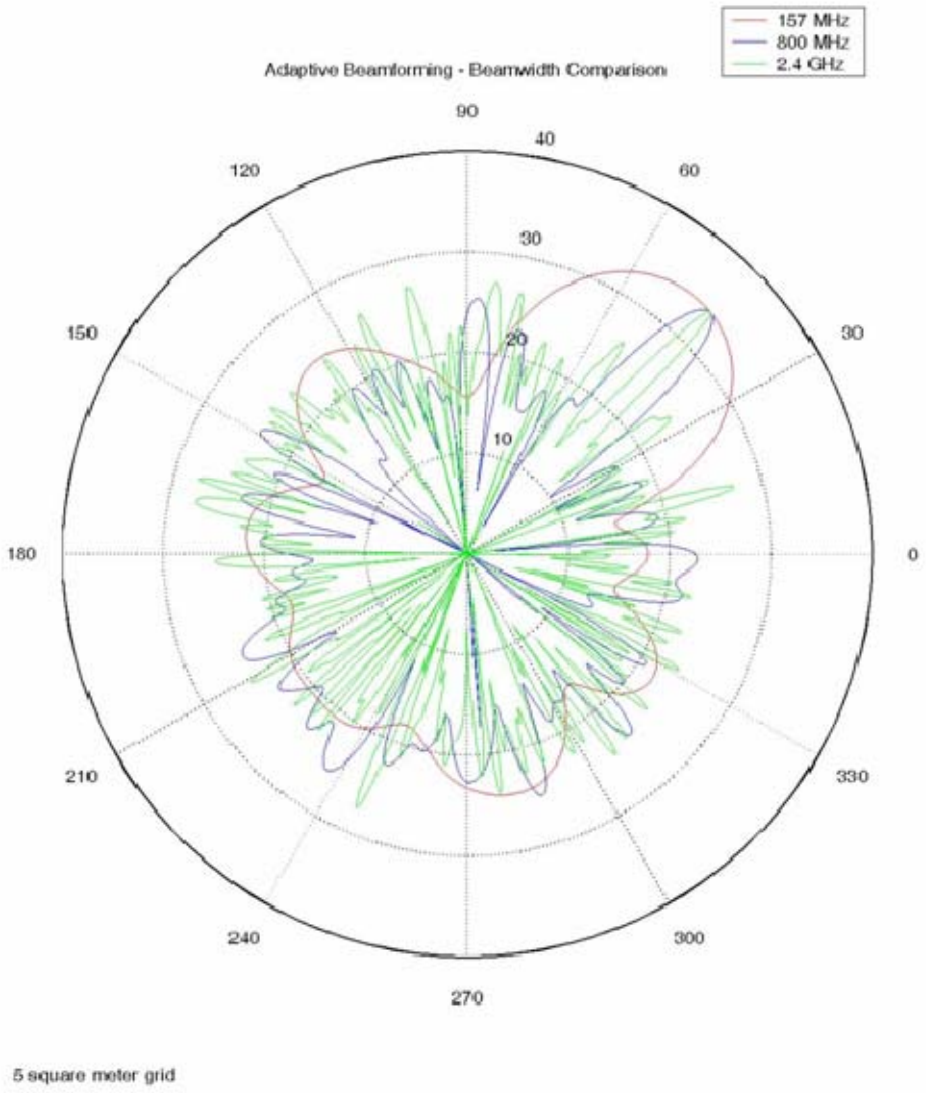


Figure 99. Array Pattern determined utilizing Method One with L_1 = 50 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As can be seen from the figures, the gain of the mainbeam went from 25 dB for $L_1=10$ iterations to 35 dB for $L_1=50$ iterations. Unfortunately, it can be seen that this method is not capable of mitigating the grating lobes. As the iterations are increased the

sidelobes have continued to increase and the mainlobe has continued to decrease. Therefore this method would not be capable of performing direction finding.

Now the area, A_2 will be increased to 400 m^2 as shown in Figure 100.

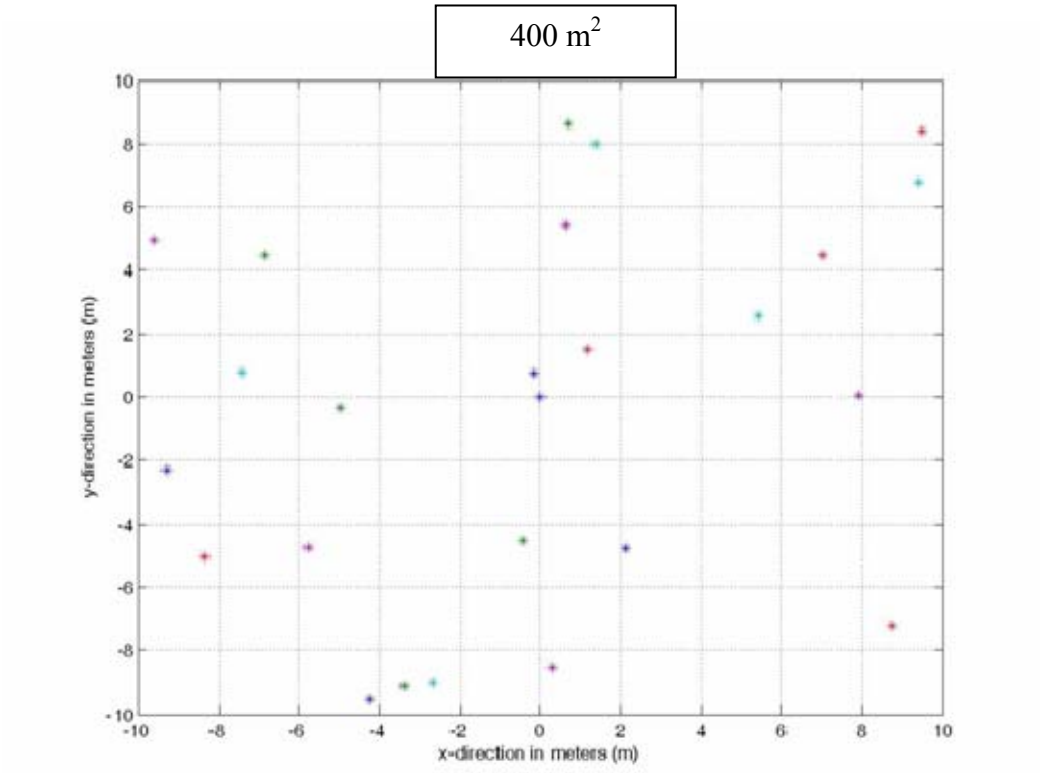


Figure 100. 25 Node array at a node density of 1.25 sensors/m^2.

Freq (MHz)	Noise Power (pW)	Area, A_2 (m^2)	Sample Size	# of iterations L_1	Number of Sensors per subset, P	Node Density (sensors/ m^2)
157/800/2400	1	400	500	10	25	0.0625
157/800/2400	1	400	500	20	25	0.0625
157/800/2400	1	400	500	50	25	0.0625

Table 19. Adaptive beamforming utilizing Method One with a node density of 0.0625 sensor/m^2.

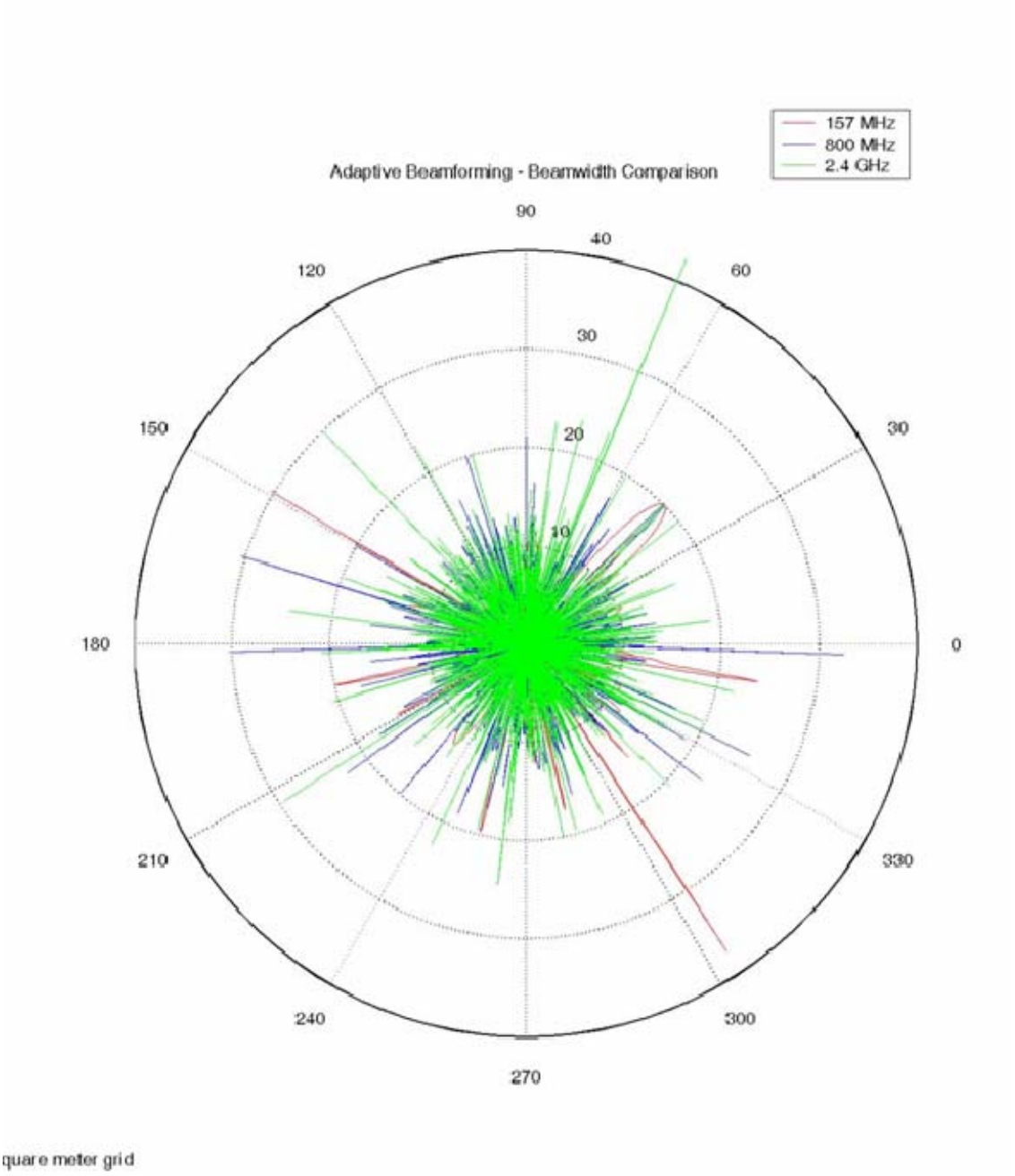


Figure 101. Array Pattern determined utilizing Method One with L_1 = 10 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

Once again the side lobes are present at 800 MHz and 2400 MHz. Also it is worth noting that the solution has larger sidelobes than in the previous figures. This intuitively makes sense because with more distance the effect of the grating lobes becomes more prominent.

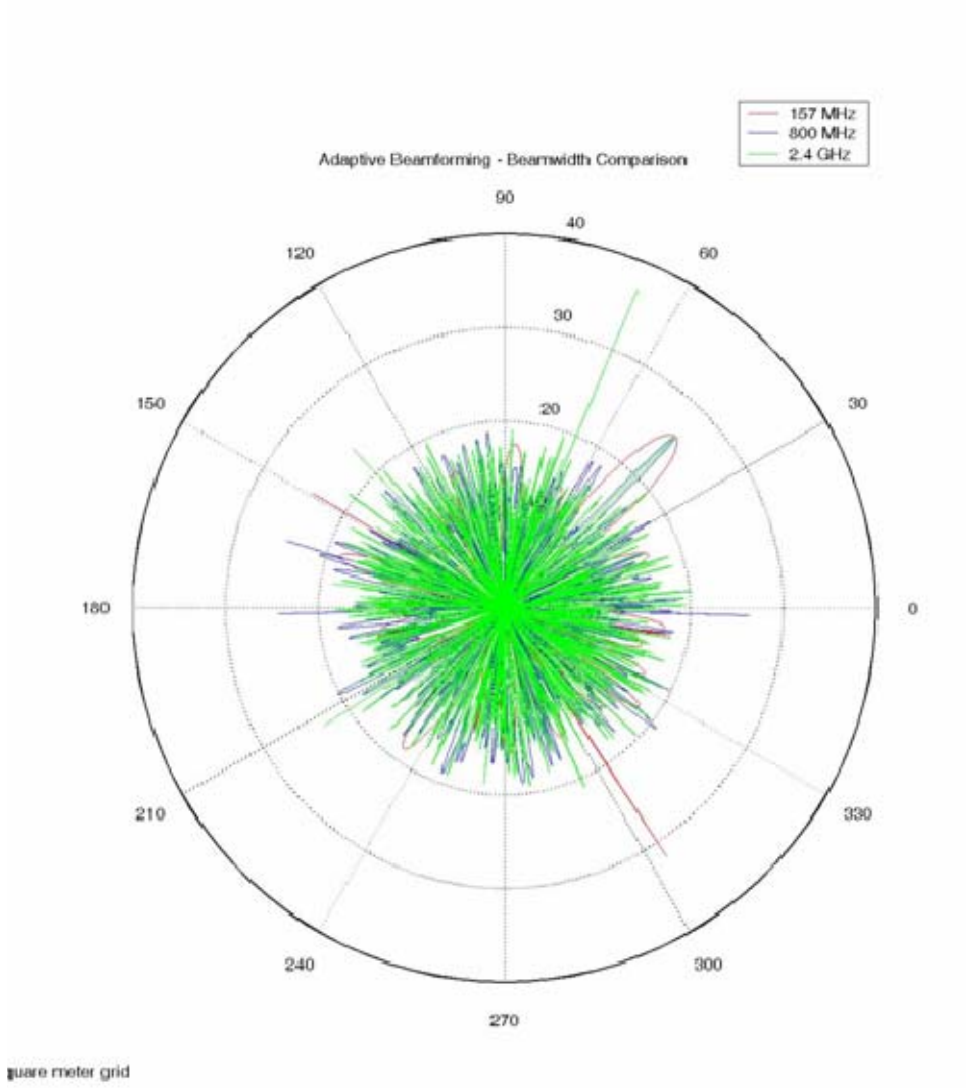


Figure 102. Array Pattern determined utilizing Method One with $L_1 = 20$ iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

Increasing the number of iterations has not reduced the side lobes at 800 MHz and 2400 MHz. The mainbeam has increased to 25 dB, but the beamwidth is getting wider as the side lobes increase.

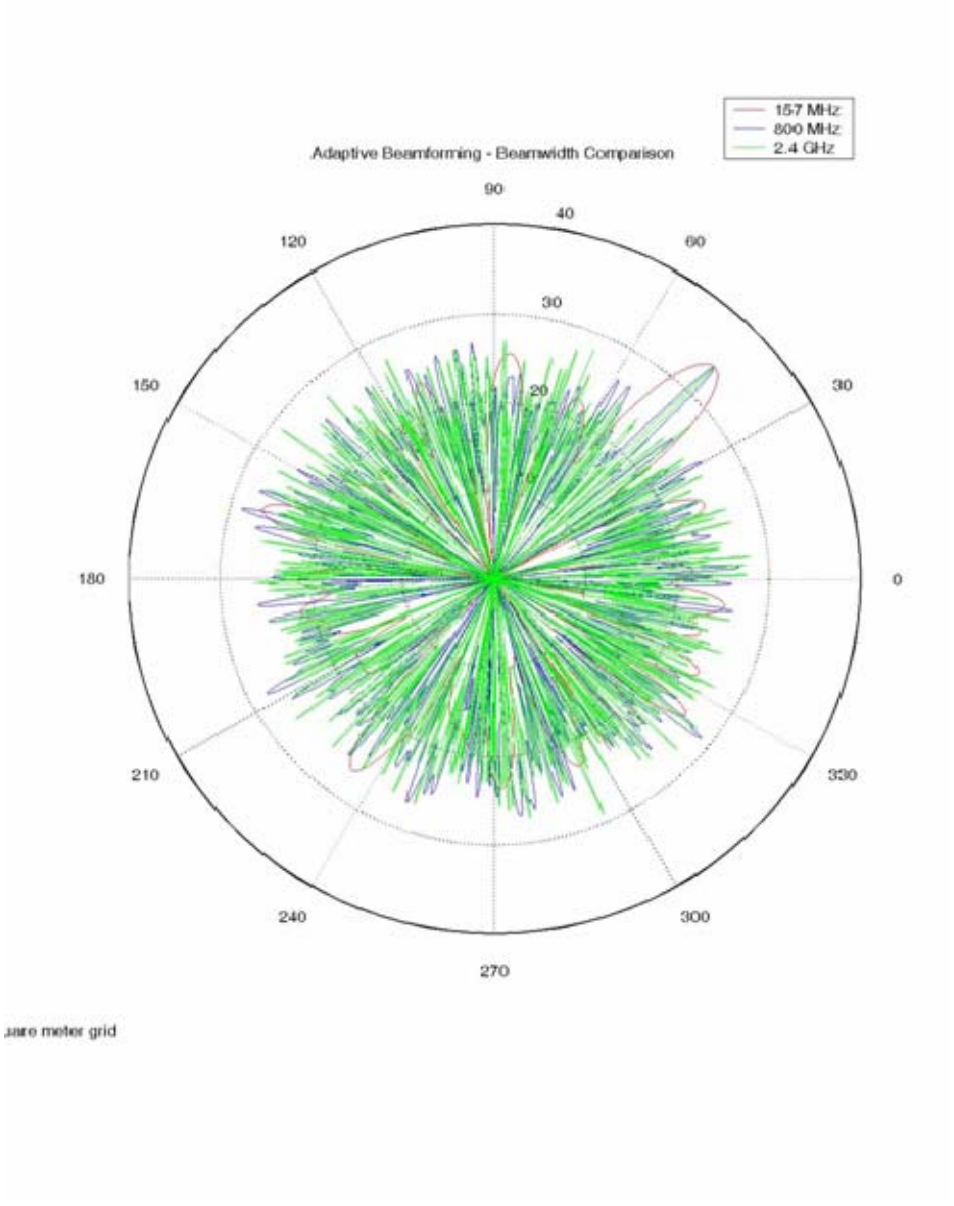


Figure 103. Array Pattern determined utilizing Method One with L_1 = 50 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

The mainbeam is now at 35 dB and the sidelobes have increased, which is similar to the case for the 25 square meter grid. Also the mainbeam beamwidth has continued to get wider.

Next the area, A_2, was increased to 2500 m^2 as shown in Figure 104.

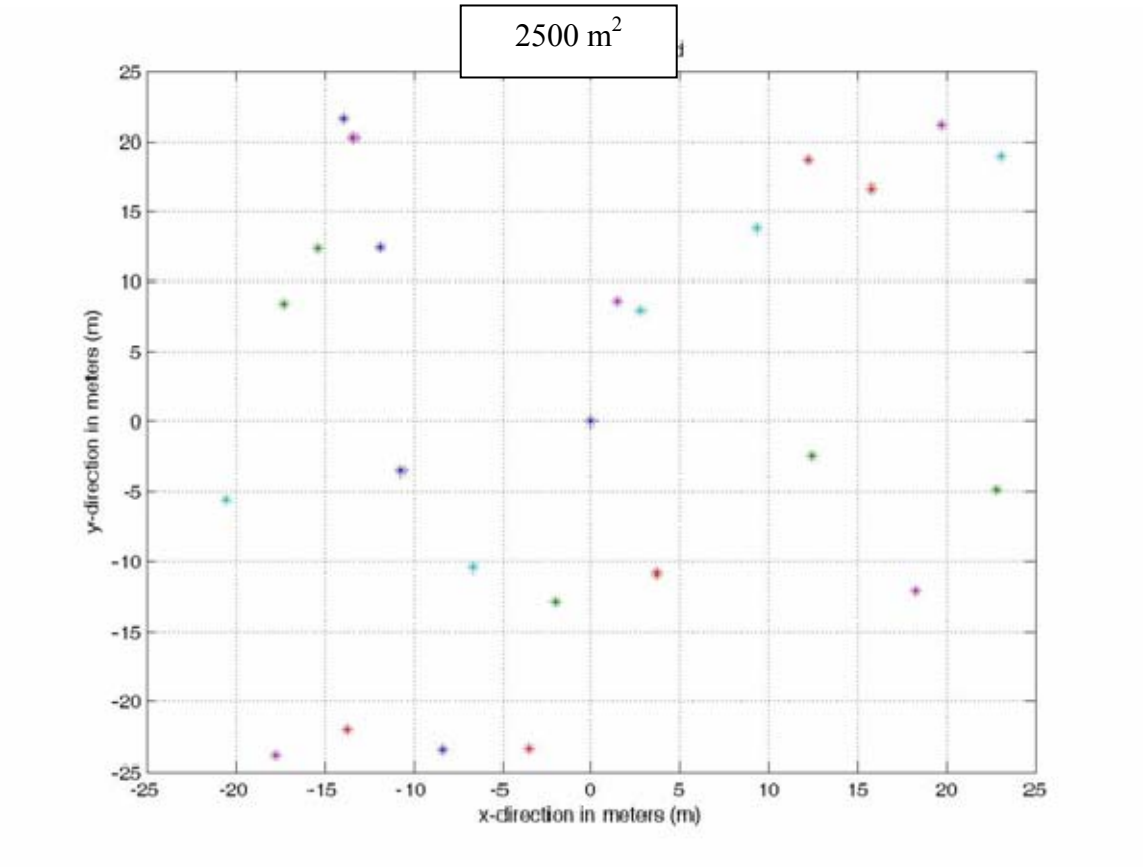


Figure 104. 25 Node array at a node density of 0.01 sensors/m^2.

Freq (MHz)	Noise Power (pW)	Area, A_2 (m^2)	Sample Size	# of iterations L_1	Number of Sensors per subset, P	Node Density (sensors/m^2)
157	1	2500	500	10	25	0.01
800	1	2500	500	20	25	0.01
2400	1	2500	500	50	25	0.01

Table 20. Adaptive beamforming utilizing Method One with a node density of 0.01 sensor/m^2.

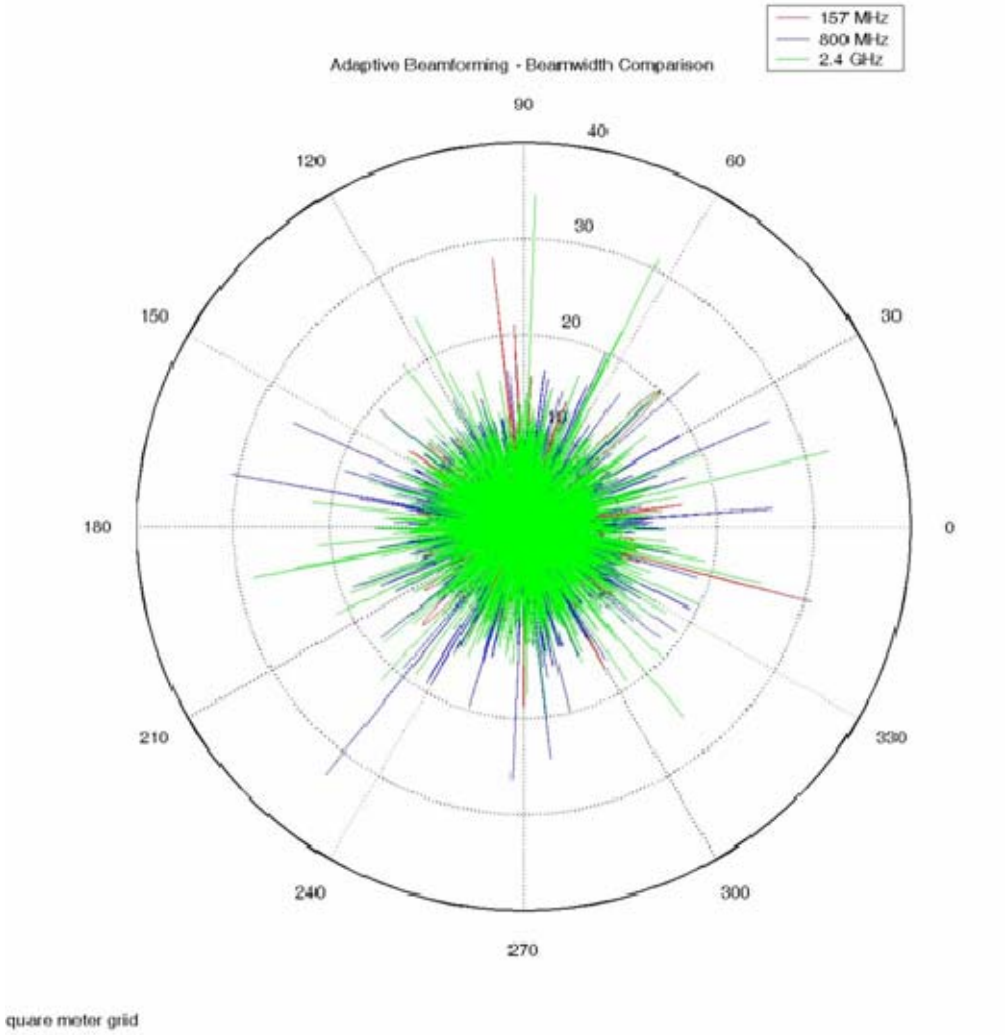


Figure 105. Array Pattern determined utilizing Method One with L_1 = 10 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

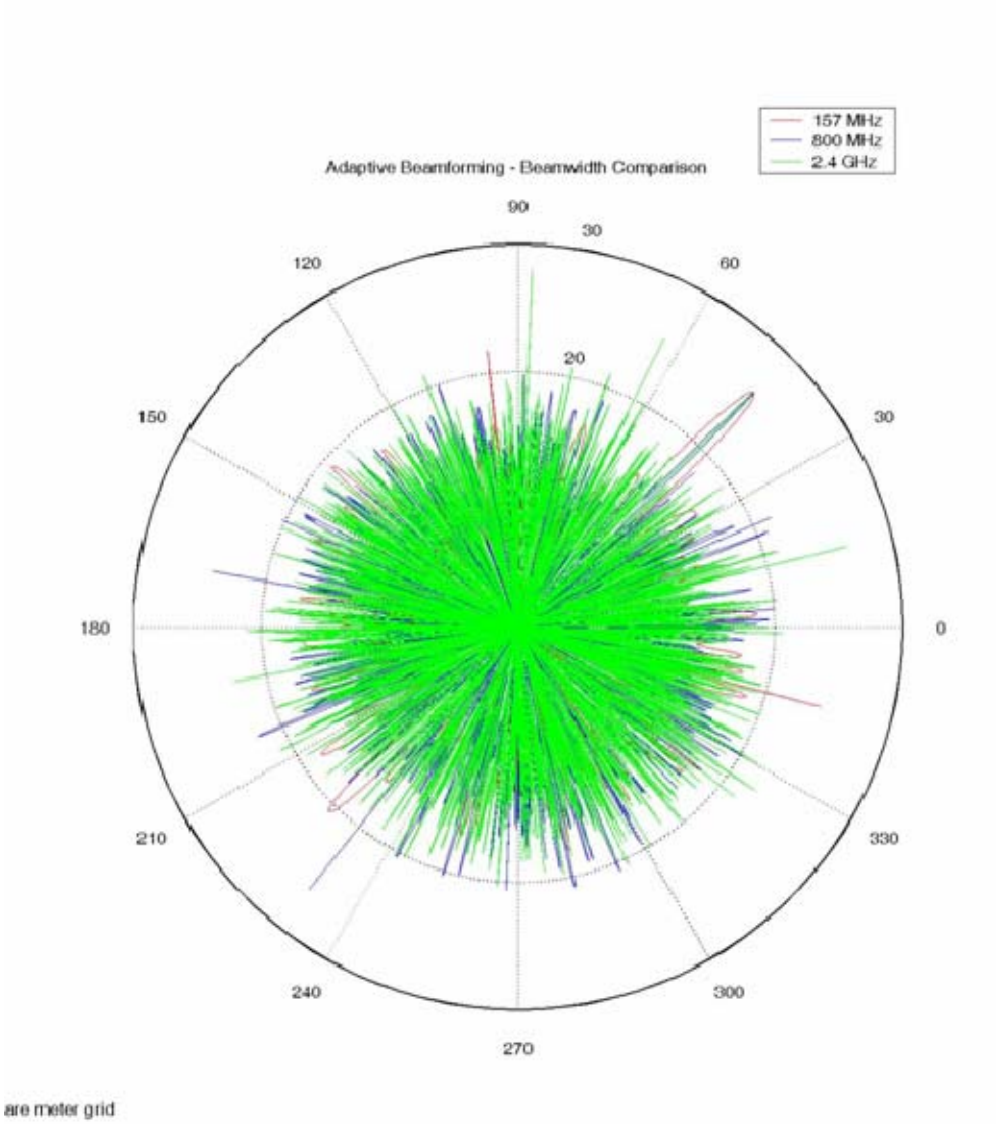


Figure 106. Array Pattern determined utilizing Method One with $L_1 = 20$ iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

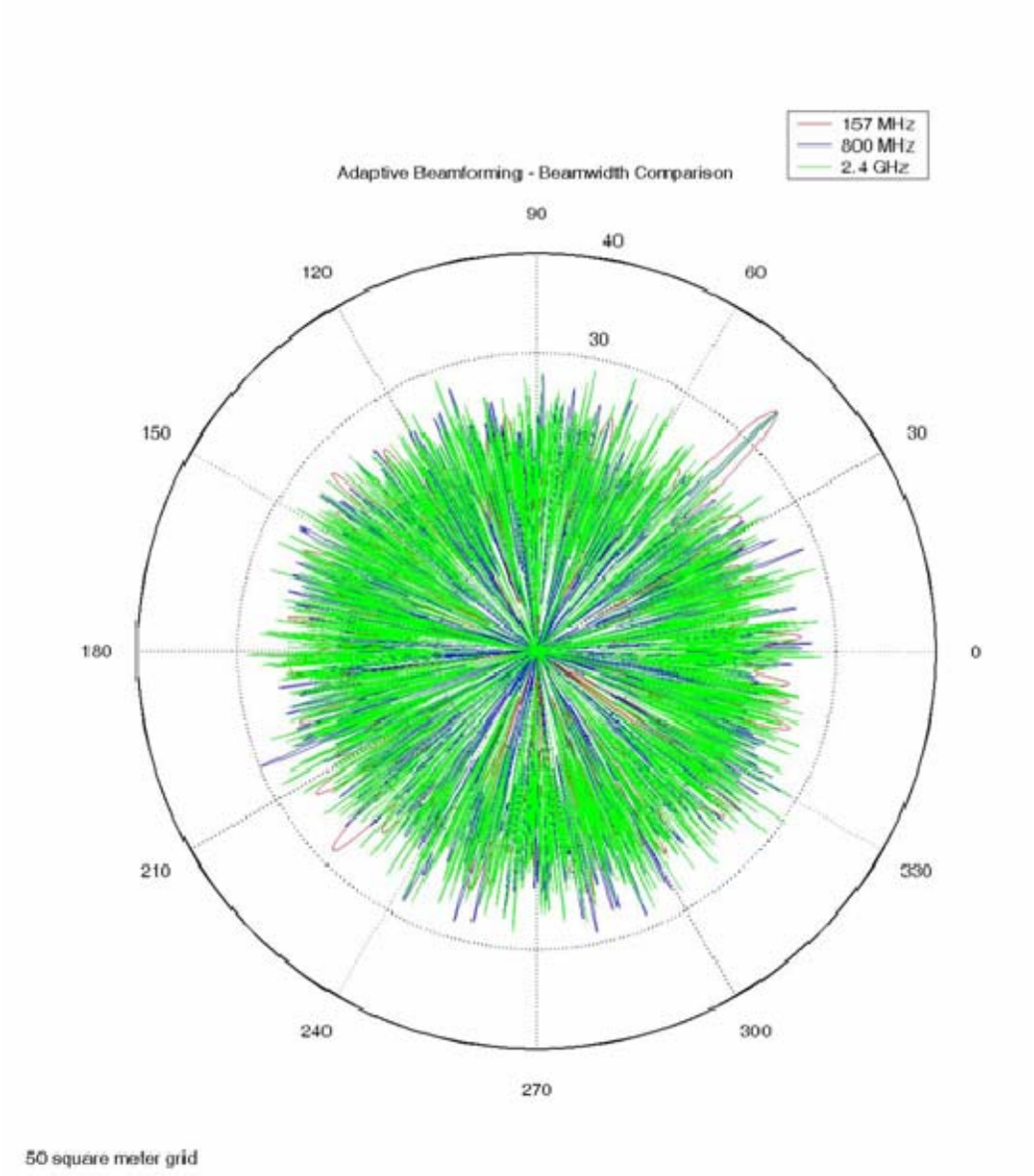


Figure 107. Array Pattern determined utilizing Method One with L_1 = 50 iterations for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

The results for the 2500 square meter grid are analogous to the ones presented for the 400 square meter grid. Side lobes spikes are still present for 800 MHz and 2400 MHz at the 10 iterations solution. As the number of iterations increased the mainbeam went from 25 dB to 35 dB. The solution is noisier because the affect of the grating lobes has

become more of a problem as the distance between sensors increased. The sidelobes rise as the number of iterations increases. The beamwidth of the mainbeam is also getting wider. These last two facts are a direct result of the grating lobe problem. The algorithm is no longer selecting randomly from the entire sensor field therefore the build-up of the periodicities on the chosen elements becomes more prevalent as the number of iterations increases.

Therefore it may be concluded that this approach is not adequate for direction finding particularly in a highly dense environment. In fact as more iterations are calculated the worse the direction finding performance becomes. Specifically, as the number of iterations increase the higher the side lobes become and the wider the mainbeam beamwidth becomes which means less directivity. In other words this methodology is not capable of mitigating the grating lobes. The amount of interference received from these side lobes would make it very difficult to localize a target of interest.

In addition, the computational burden on the central controller and the energy burden on the individual sensors would be significant. For example, where total transmissions are defined as

$$T_{\Psi} = L_1 P \,, \tag{99}$$

where L_1 is the total number of iterations required for Method One to form the desired beam and P is the number of sensors involved in the process. Therefore, P is the number of transmissions per iteration in order to obtain a solution. If the cost per transmission is δ *joules* / *transmission* , χ_1 *joules* is defined as the energy expended waiting to transmit. As stated before a node awaits for all other nodes to transmit prior to it transmitting. For convience, χ_1 is defined as a fraction of the energy to transmit the signal data. χ_2 *joules* is the energy expended to wake a node and assumes the worst case that all the nodes are in a sleep mode. For convience, χ_2 is also defined as a fraction of the energy to transmit the signal data. The total energy cost to form one beam in a given direction is defined as

$$\begin{aligned} \Gamma_{BF} &= \delta T_{\Psi} + P\chi_1 + P\chi_2 \\ &= \delta\left(L_1 P\right) + \left(P-1\right)\left(0.1\delta\right) + P\left(0.5\delta\right) \end{aligned}, \tag{100}$$

where Γ is in units of *joules*. For this method the energy cost is distributed over the same P subset of sensors resulting in an energy cost per sensor of

$$\Xi = \delta L_1 + (0.1\delta) - \frac{0.1\delta}{P} + (0.5\delta), \tag{101}$$

where Ξ is measured in units of *joules/sensor*.

Note two of the examples discussed earlier where the data is shown in Table 21. Three different frequencies were chosen for comparison, and the node densities were all the same. In all cases the subset size was *P=25* nodes.

Freq (MHz)	**# of Trans., T_Ψ**	**Area, A_2 (m^2)**	**Total Energy, Γ_Ψ (joules)**	**# of Iter., L_1**	**Energy per sensor, Ξ (j/sensor)**
157 /800 /2400	500	25	514.9δ	20	20.596δ
157 /800 /2400	1250	25	1264.9δ	50	50.596δ

Table 21. Total number of transmissions and energy expended utilizing Method One.

As can be seen from Equation (99) that as L_1 increases so does the total number of transmissions, therefore increasing the total energy use per sensor as seen from Equation (101). For example, an increase of L_1 from 20 to 50 increased the total transmissions T_Ψ from 500 transmissions to 1250 transmissions. That is an increase of total energy cost of 750δ *joules*. To make matters worse increasing the number of transmissions did not mitigate the grating lobes. Therefore this method is not suitable for beamforming and is especially not suited for direction finding.

2. Method Two

Now it will be shown that randomly selecting a different *P* subset of nodes per iteration from a large SIGINT array produces much better results. Using Equation (96) the central controller takes another set of data samples from a new subset of *P* nodes. Then the process described earlier is repeated for L_2 iterations where L_2 is used to

represent Method Two. The final result is added to the previously stored result. This magnitude is then plotted in dB against all possible values of azimuth and elevation. This process is repeated until a desired solution has been found.

Again the nodes were randomly distributed within a square grid. The number of *M* elements equals the number of *N* elements to facilitate easier simulations. The software taken from [72] was modified to create the effect of the central controller selecting data from a different *P* subset of nodes for each iteration. This modified code can be found in Appendix I. Above, the node density was varied by having the central controller select sensors that were distributed over different areas, A_2. Another way to change the node density is by having more sensors, Λ, placed within a given area, A_2, and was investigated. For an array containing $\Lambda \geq 75$ sensors the area, A_2, was changed between 25, 2500 and 250000 m^2 to characterize the effects on the beamforming process.

Freq (MHz)	**Noise Power (pW)**	**Area, A_2 (m^2)**	**Sample Size**	**# of iterations L_2**	**Number of Sensors per subset, P**	**Node Density (sensors/m^2)**
157/800/2400	1	25	500	3	≥75	≥3
157/800/2400	1	2500	500	3	≥75	≥0.03
157/800/2400	1	250000	500	3	≥75	≥0.0003

Table 22. Adaptive beamforming utilizing Method Two with varying node selection.

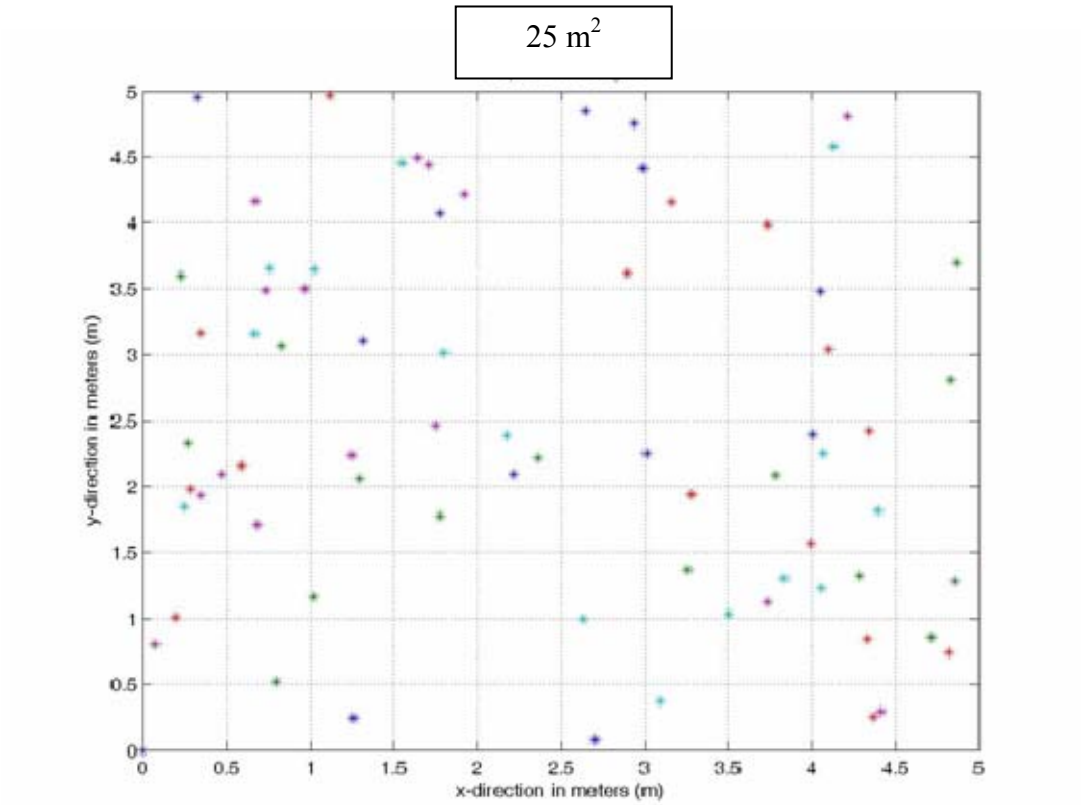
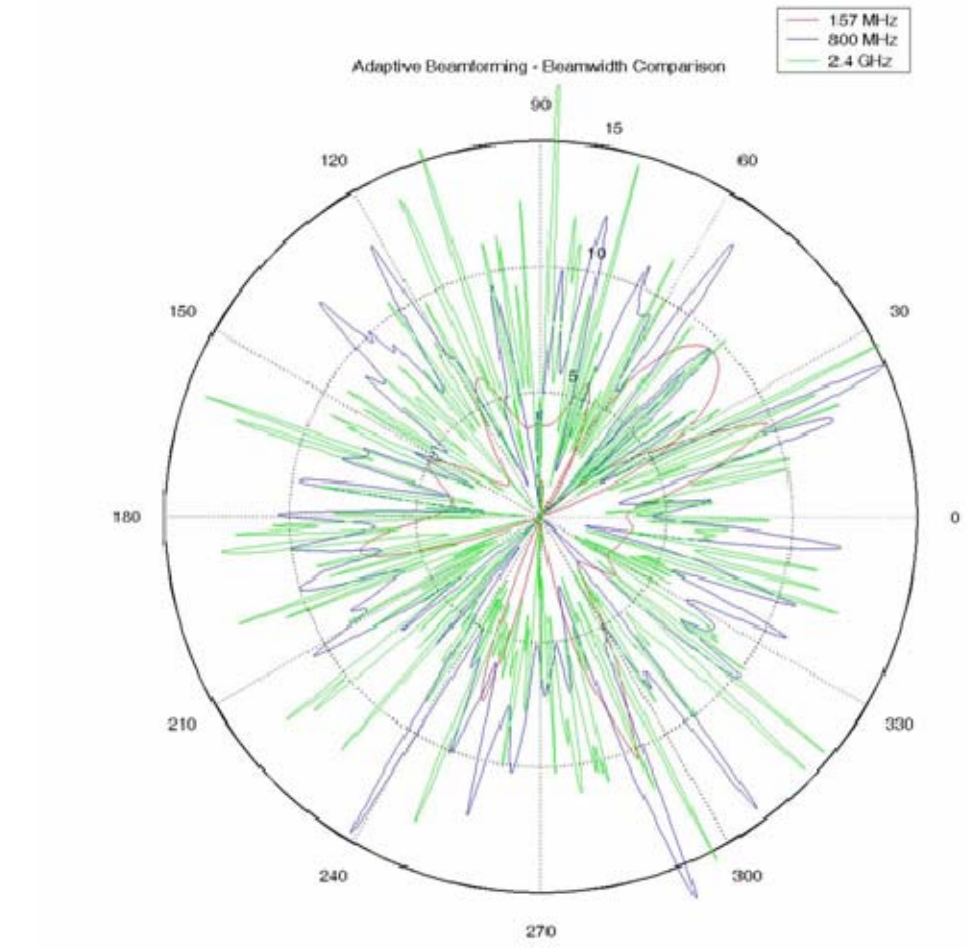


Figure 108. 75 randomly chosen sensors from a node array with a node density ≥ 3 sensors/m^2.



25 m^2

Figure 109. Array Pattern for a node density of ≥3 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As can be seen in the plots the beamwidth gets smaller as the frequency goes up. Also present at the higher frequencies are a lot of sidelobe spikes that make the polar plot unreadable. This node density does not appear to provide satisfactory results.

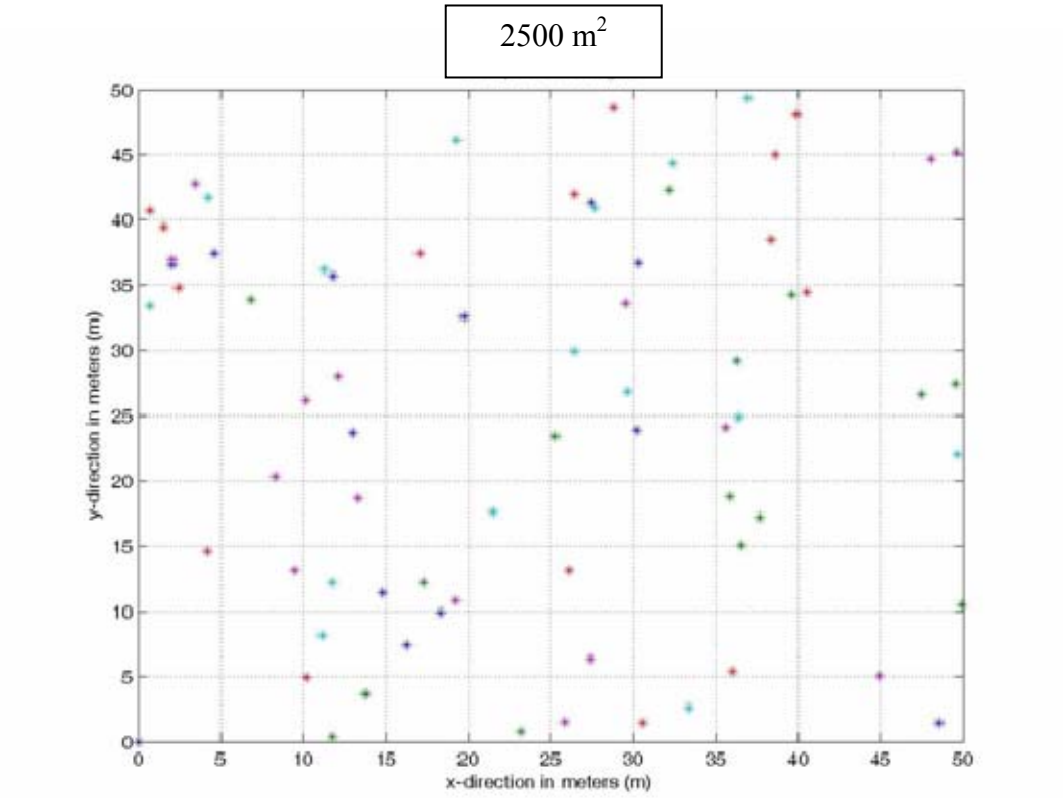


Figure 110. 75 randomly chosen sensors from a node array with a node density ≥ 0.03 sensors/m^2.

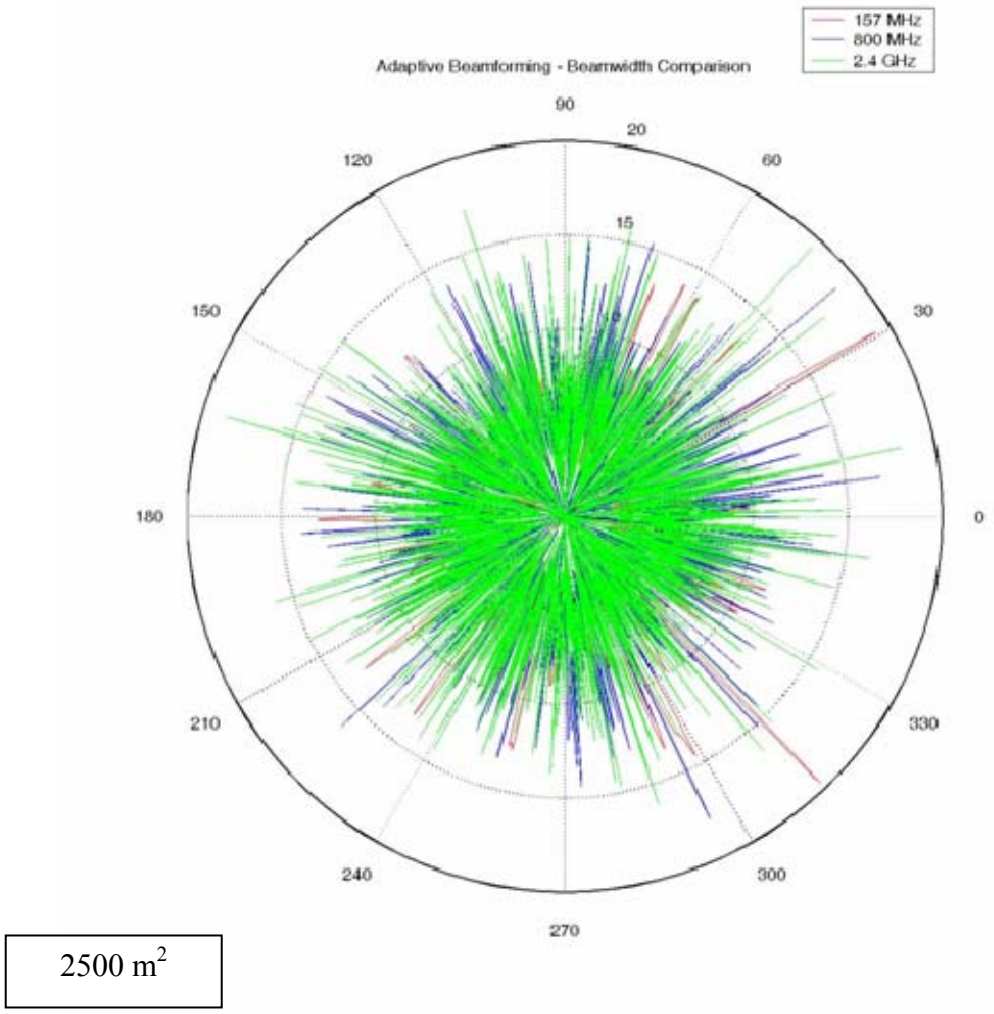


Figure 111. Array Pattern for a node density of ≥0.03 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

The sidelobes in the Figure 111 (a) make it unreadable. Upon closer observation, it is noticed that the sidelobes are much higher than in Figure 109. This can be attributed to the fact that the distance has increased between the nodes, making the grating lobes more of a problem. It is also worth noting that the beamwidth of the mainbeam narrowed considerably as expected since the node density has decreased.

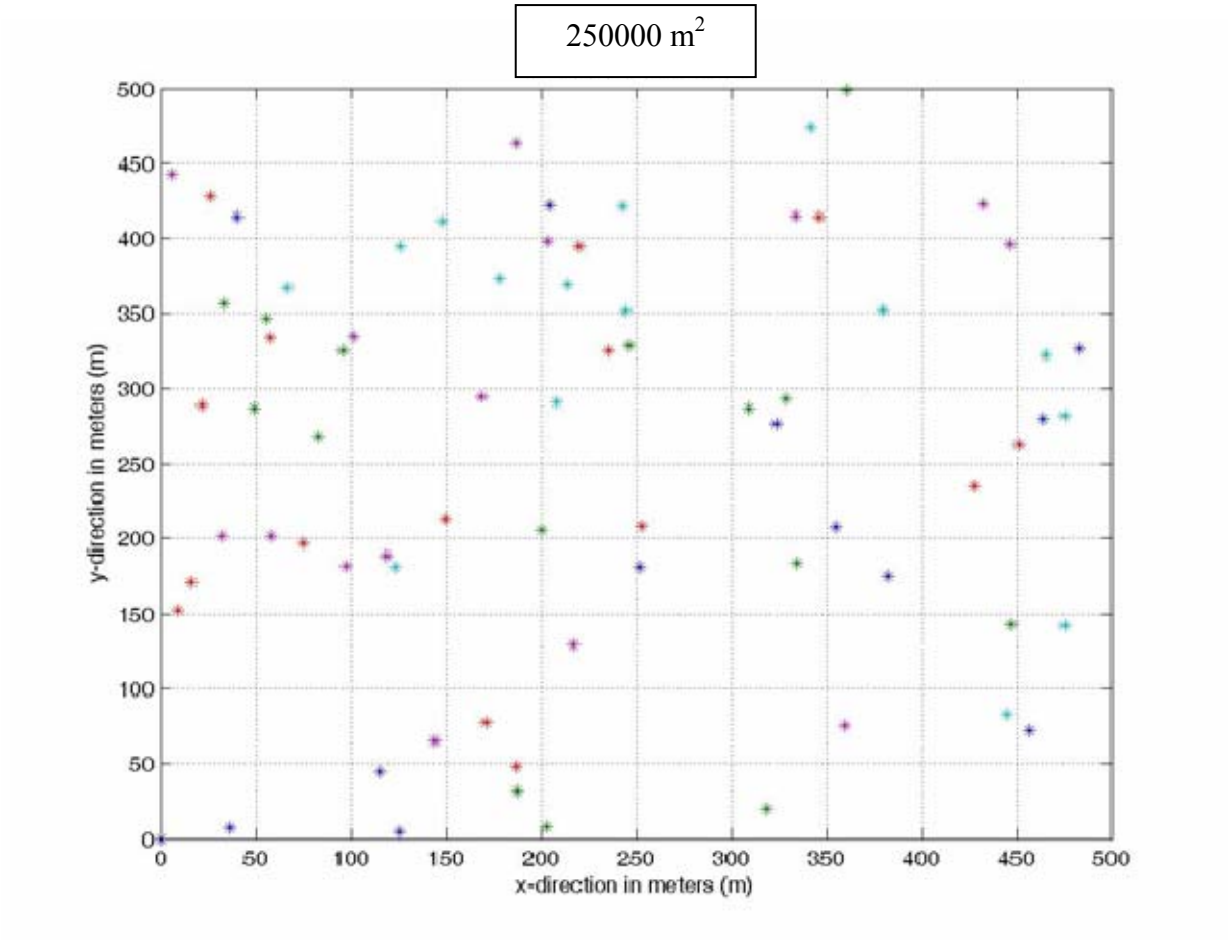


Figure 112. 75 randomly chosen sensors from a node array with a node density ≥ 0.0003 sensors/m^2.

250000 m^2

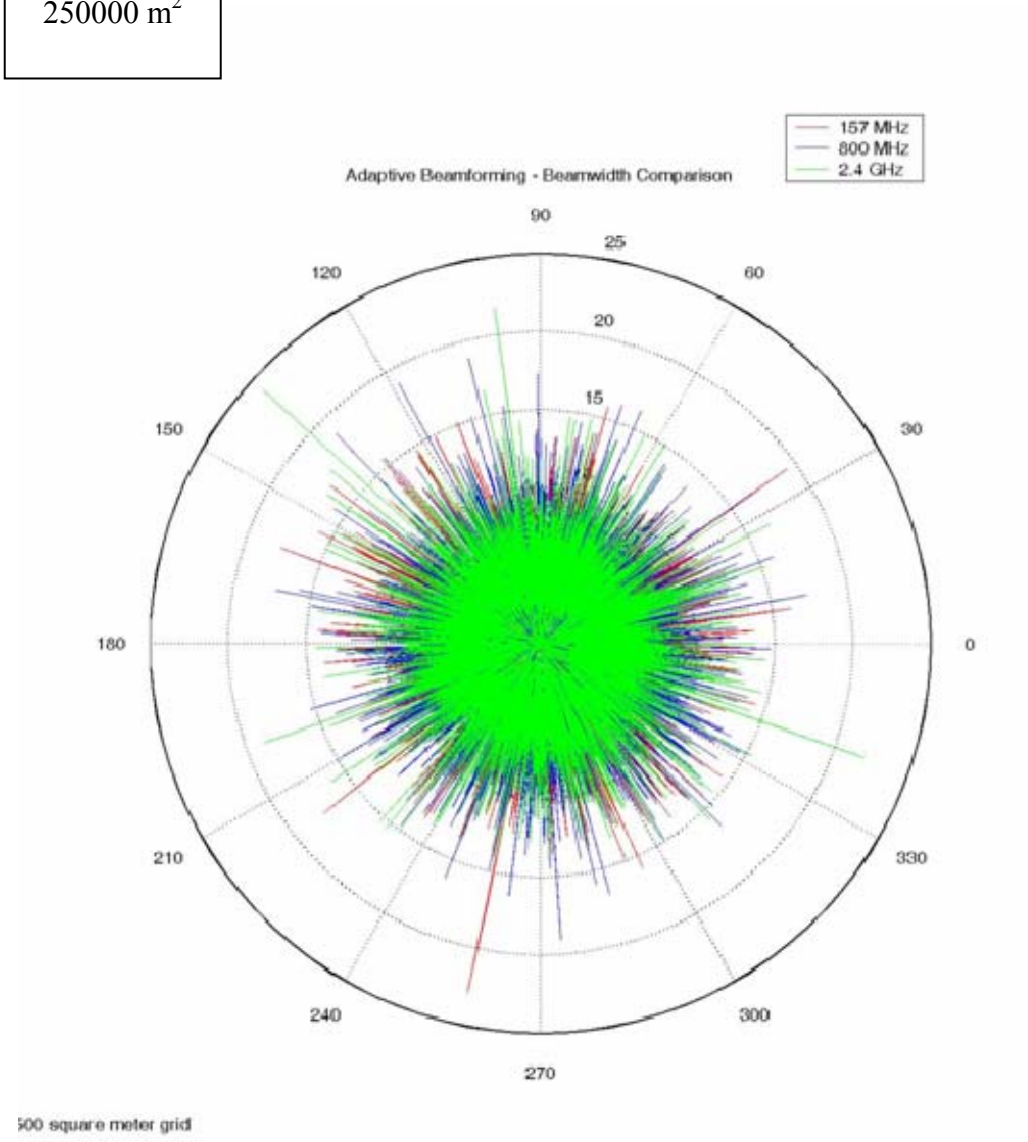


Figure 113. Array Pattern for a node density of ≥0.0003 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

The figure on the left still has too many side lobes present to be of any use. However, the figure on the right shows a very narrow beam. It can be observed that the beamwidth of the mainbeam has decreased dramatically. These results lead to the conclusion that to narrow the beam, the further apart the sensors are spaced the more

narrow the beam. A more in-depth discussion on beamwidth will be presented in subsection 4 of this chapter. Now an array size of $\Lambda \geq 125$ sensors will be investigated. Again the area, A_2, was altered between 25, 2500 and 2500000 m^2. In the next section since the overall number of sensors will be increased, this results in an increase in node densities.

Freq (MHz)	Noise Power (pW)	Area, A_2 (m^2)	Sample Size	# of iterations L_2	Number of Sensors per subset, P	Node Density (sensors/m^2)
157/800/2400	1	25	500	5	≥125	≥5
157/800/2400	1	2500	500	5	≥125	≥0.05
157/800/2400	1	250000	500	5	≥125	≥0.0005

Table 23. Adaptive beamforming utilizing Method Two with a varying node.

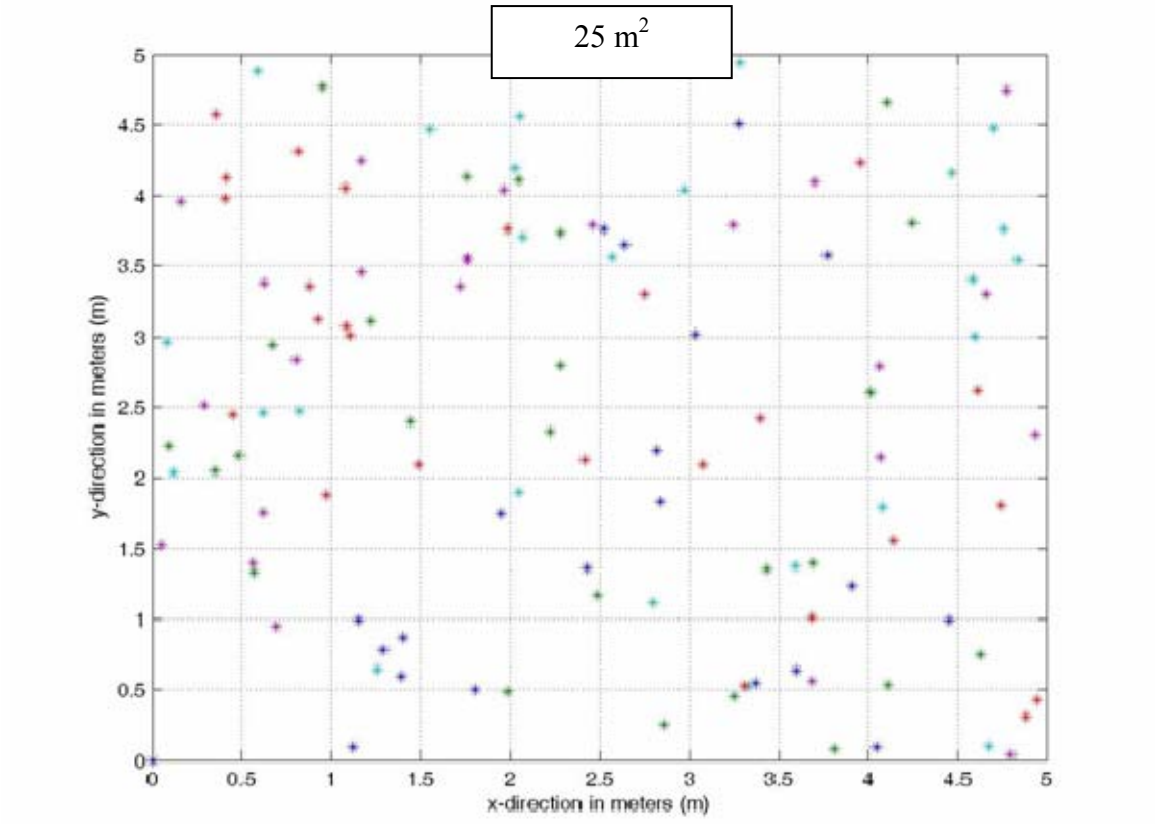


Figure 114. 125 randomly chosen sensors from a node array with a node density ≥ 5 sensors/m^2.

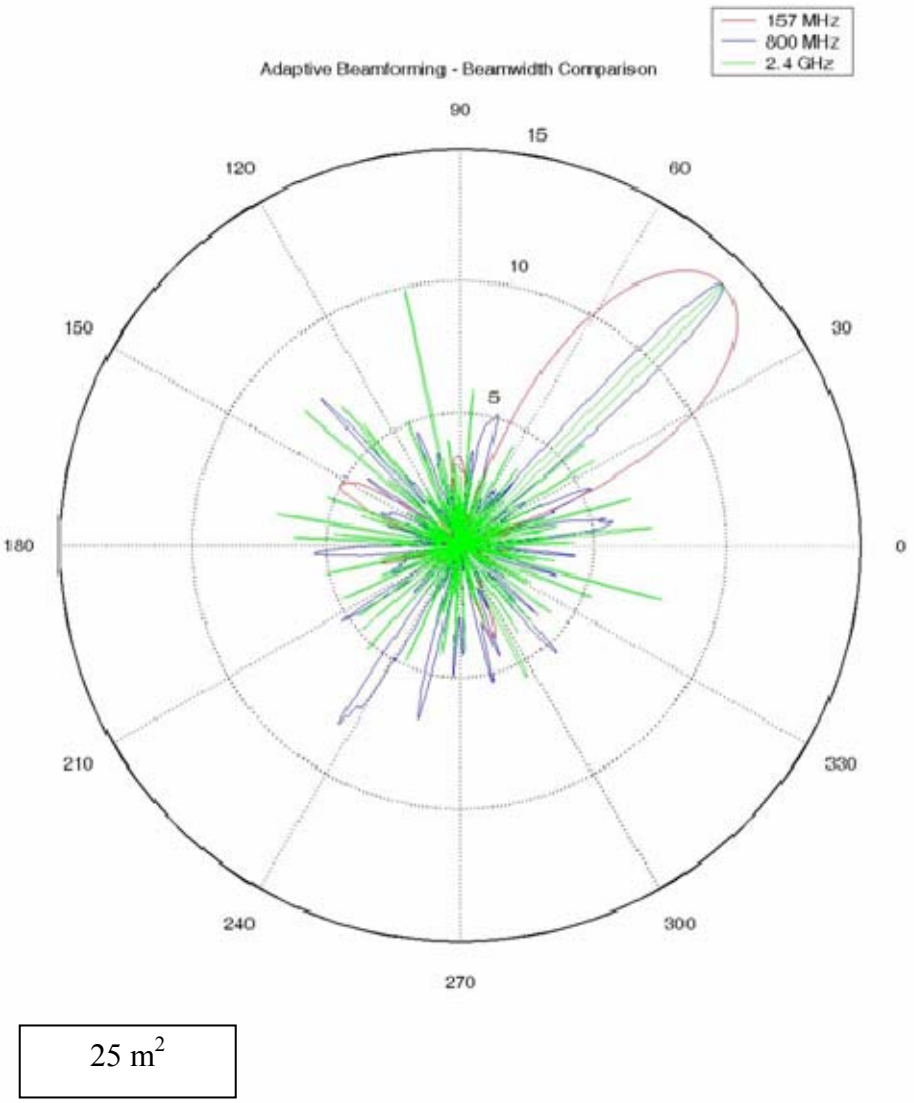


Figure 115. Array Pattern for a node density of ≥5 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

These figures show a nice beamwidth formation for all three frequencies. With a few rogue sidelobes for 800 MHz and 2400 MHz frequencies, but much less than in the previous figures.

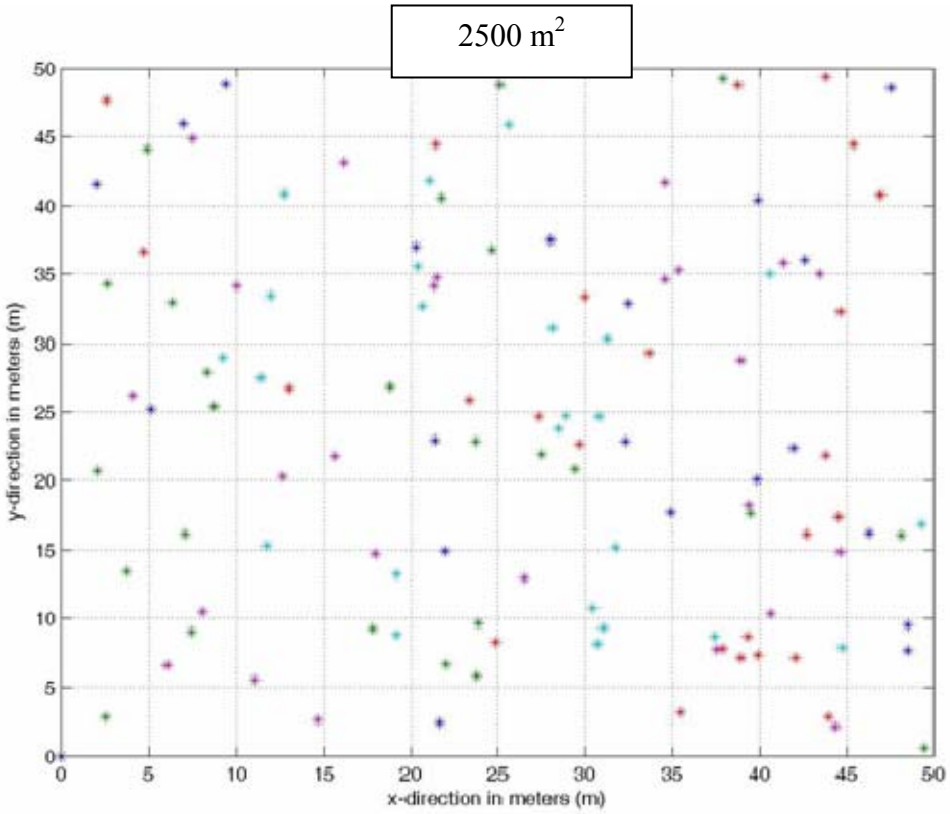


Figure 116. 125 randomly chosen sensors from a node array with a node density ≥ 0.05 sensors/m^2.

2500 m^2

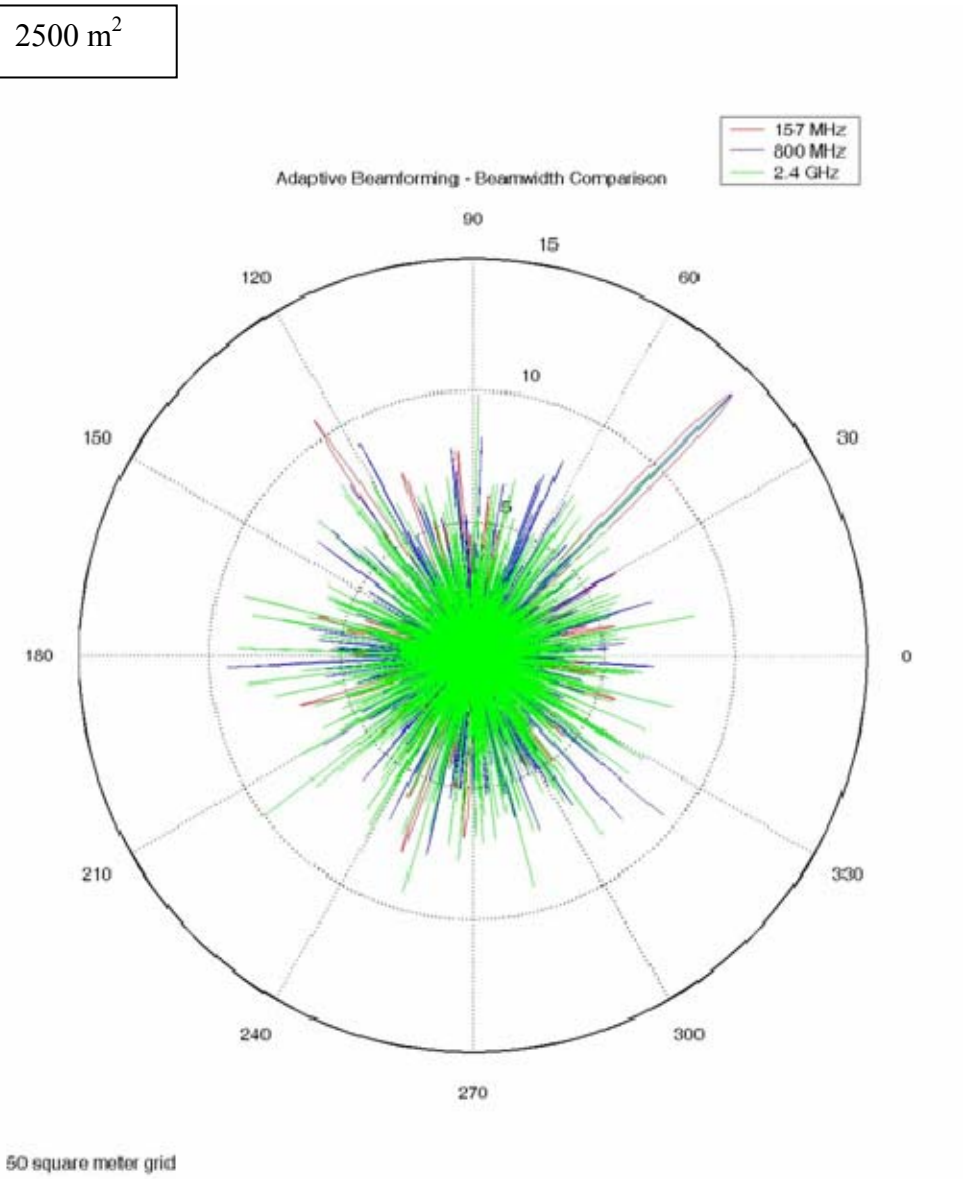


Figure 117. Array Pattern for a node density of ≥0.05 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

The results are as expected. The mainbeam beamwidth has been drastically reduced. The sidelobes appear much noisier.

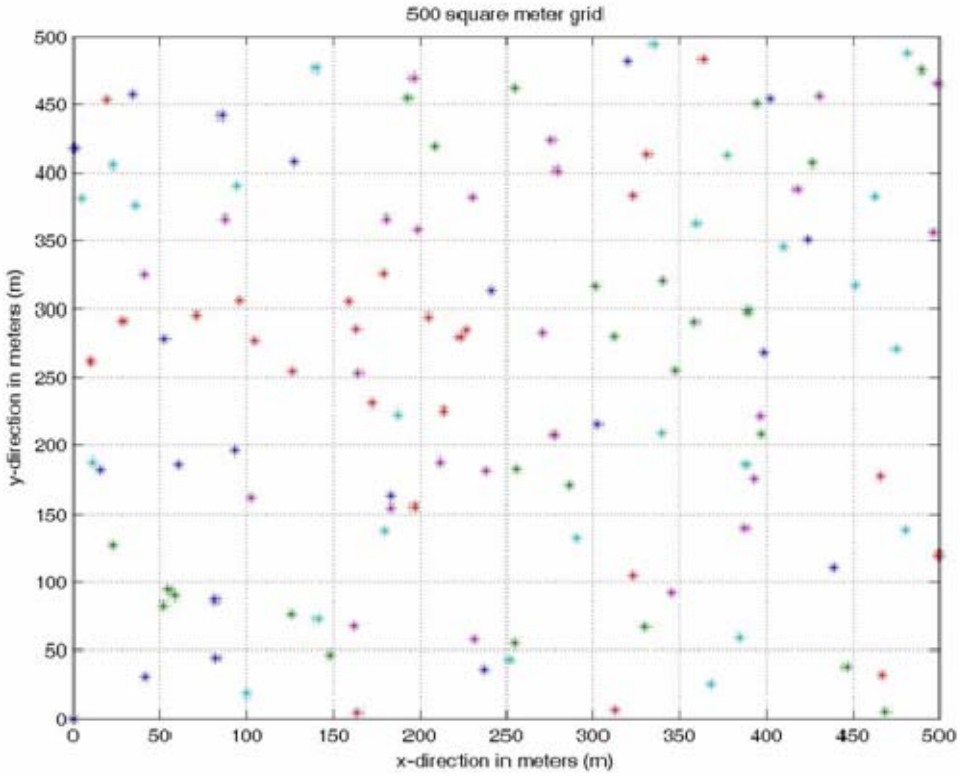


Figure 118. 125 randomly chosen sensors from a node array with a node density ≥ 0.0005 sensors/m^2.

250000 m^2

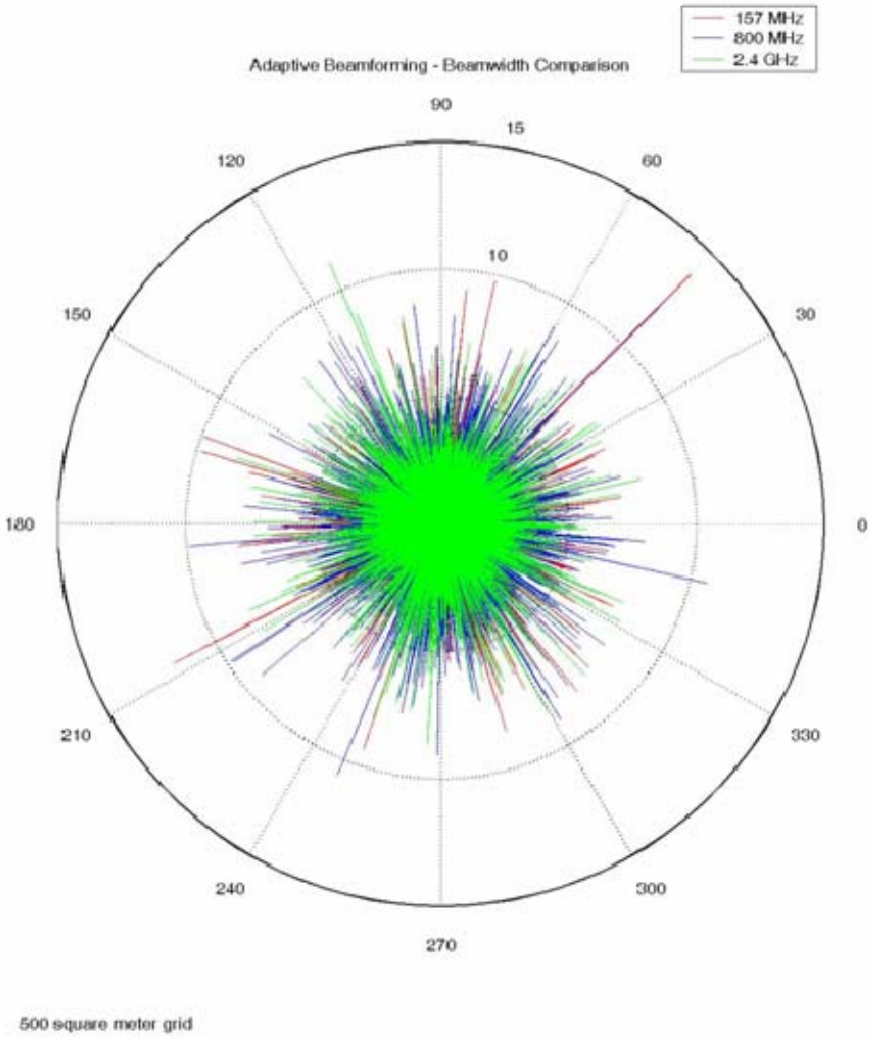


Figure 119. Array Pattern for a node density of ≥0.0005 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As can be observed from the figures, the mainbeam beamwidth has been reduced significantly. There is also an increase in sidelobes as the node density has been reduced.

Now an array size of $\Lambda \geq 175$ sensors will be investigated. The area, A_2, is altered between 25, 2500 and 2500000 m^2. This will have the affect of increasing the node densities at the respective areas.

Freq (MHz)	Noise Power (pW)	Area, A_2 (m^2)	Sample Size	# of iterations L_2	Number of Sensors per subset, P	Node Density (sensors/m^2)
157/800/2400	1	25	500	7	≥175	≥7
157/800/2400	1	2500	500	7	≥175	≥0.07
157/800/2400	1	250000	500	7	≥175	≥0.0007

Table 24. Adaptive beamforming utilizing Method Two with a varying node.

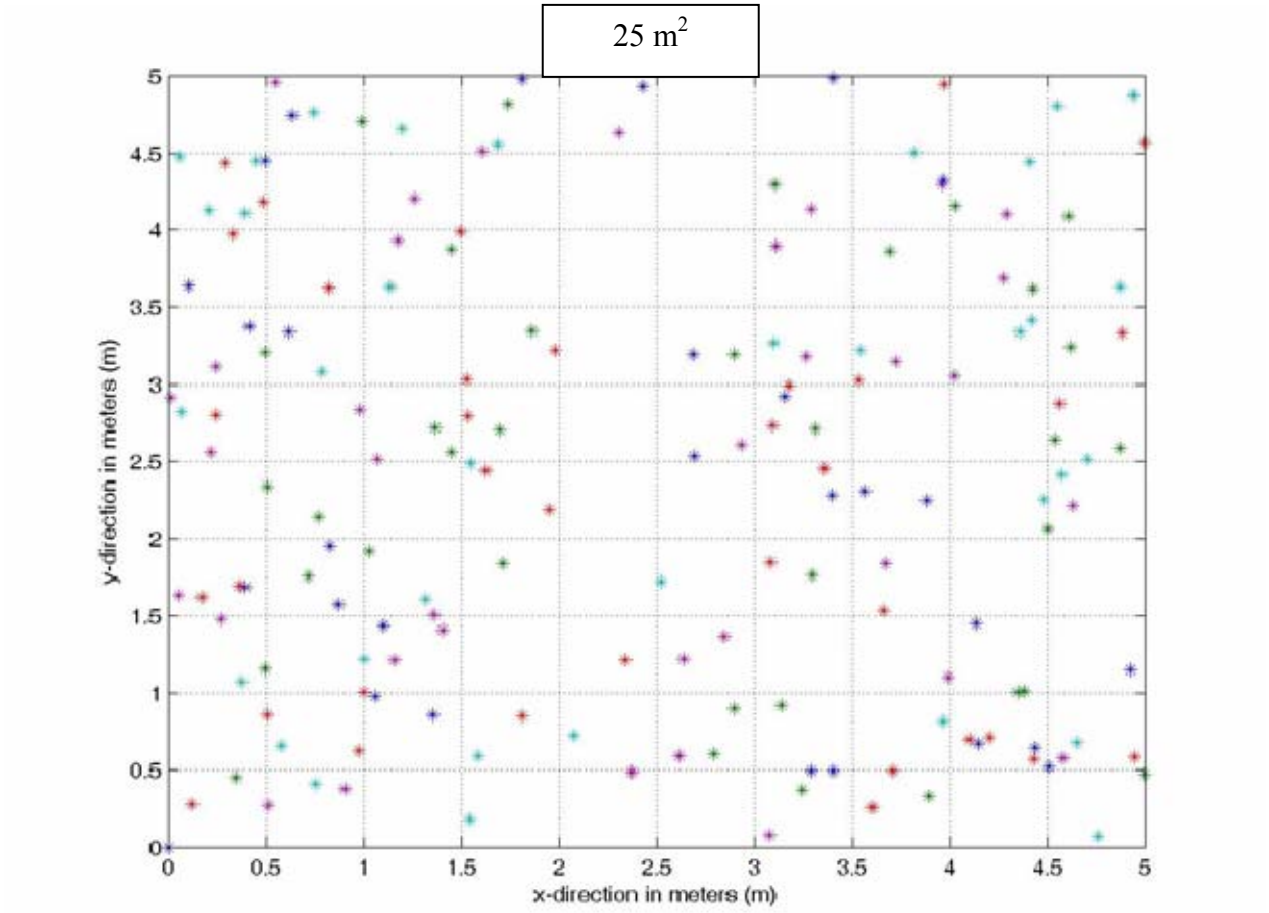


Figure 120. 175 randomly chosen sensors from a node array with a node density ≥ 7 sensors/m^2.

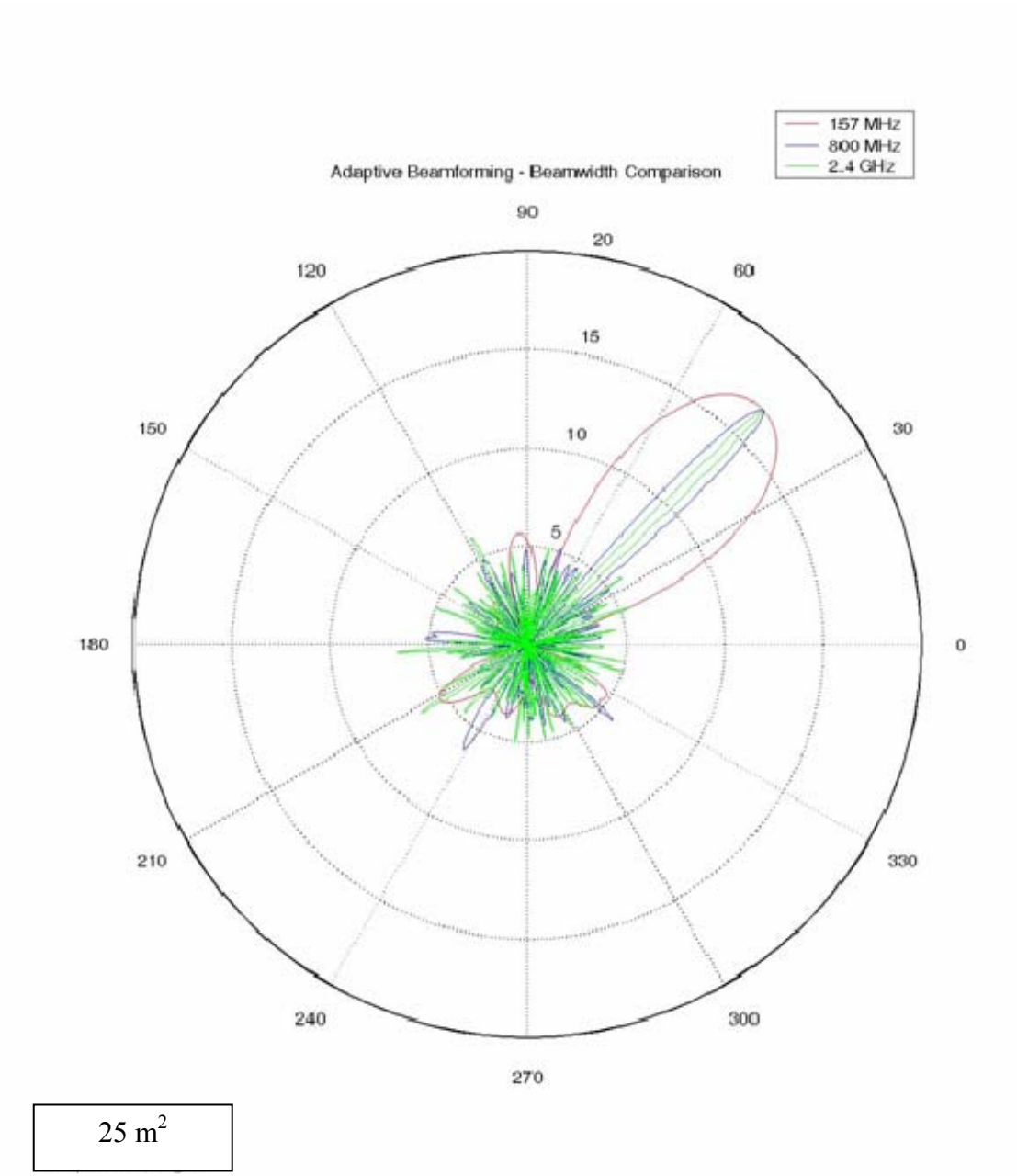


Figure 121. Array Pattern for a node density of ≥7 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

This figure shows a nice well defined beam with few rogue side lobes. As expected the mainbeam beamwidth decreases as the frequency goes up.

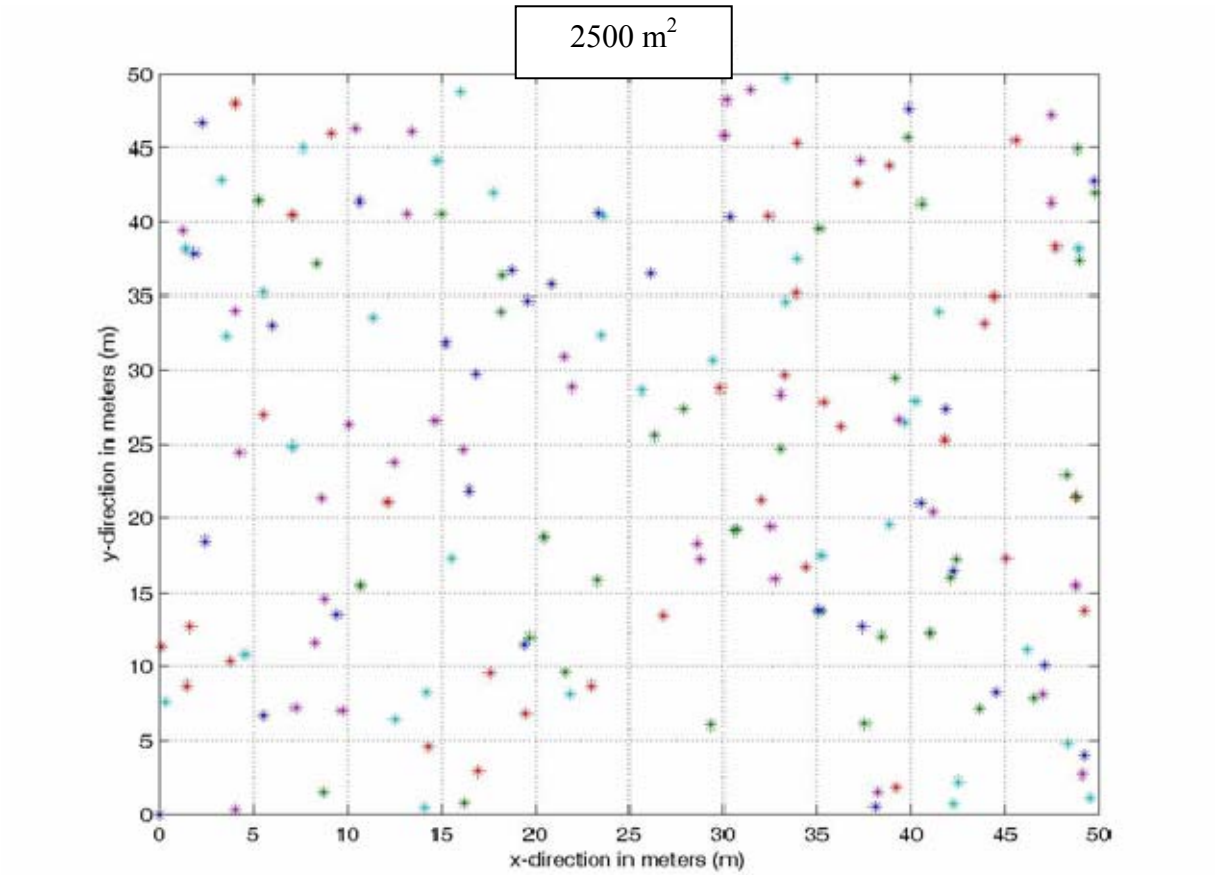


Figure 122. 175 randomly chosen sensors from a node array with a node density ≥ 0.07 sensors/m^2.

2500 m^2

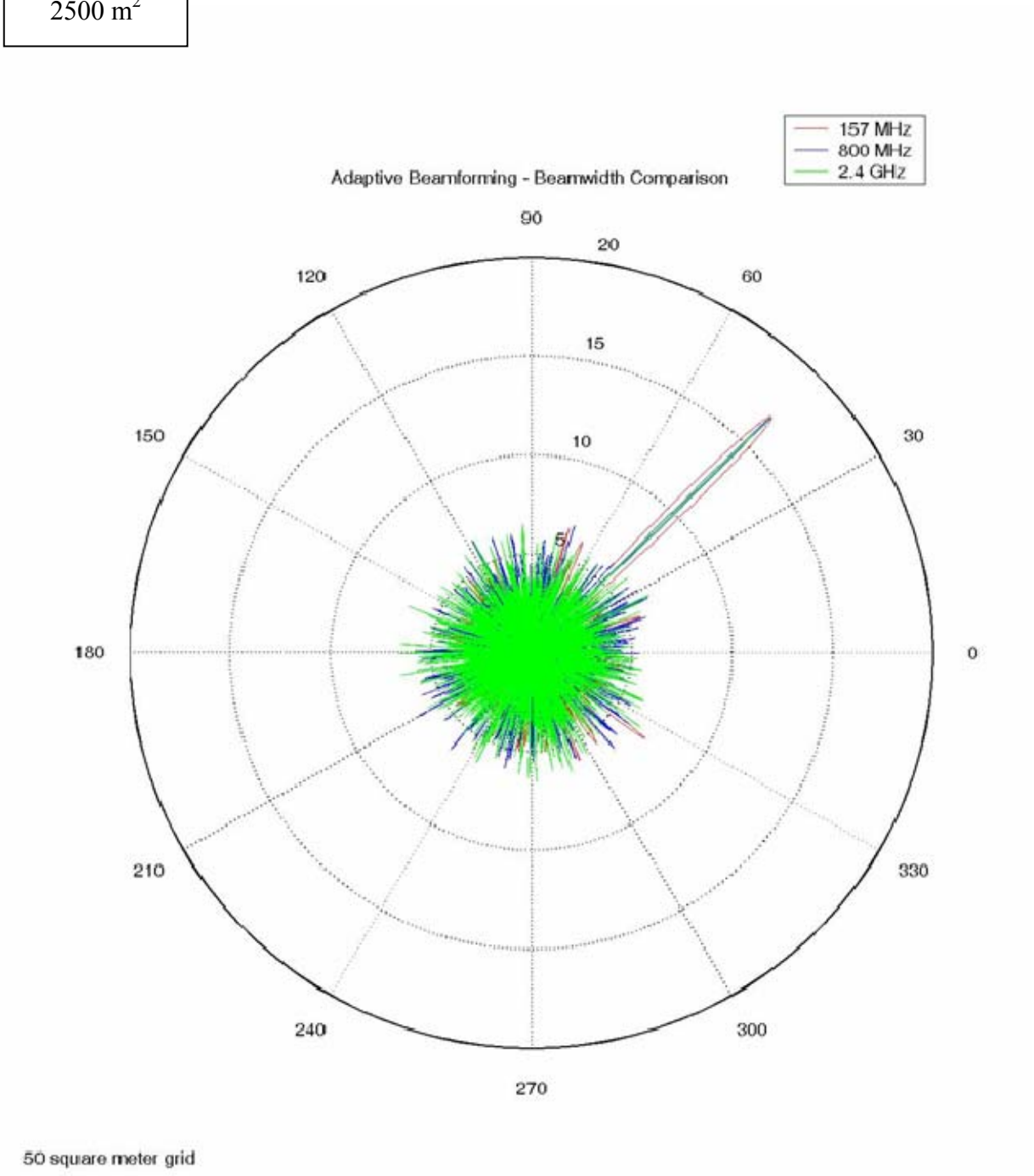


Figure 123. Array Pattern for a node density of ≥0.07 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

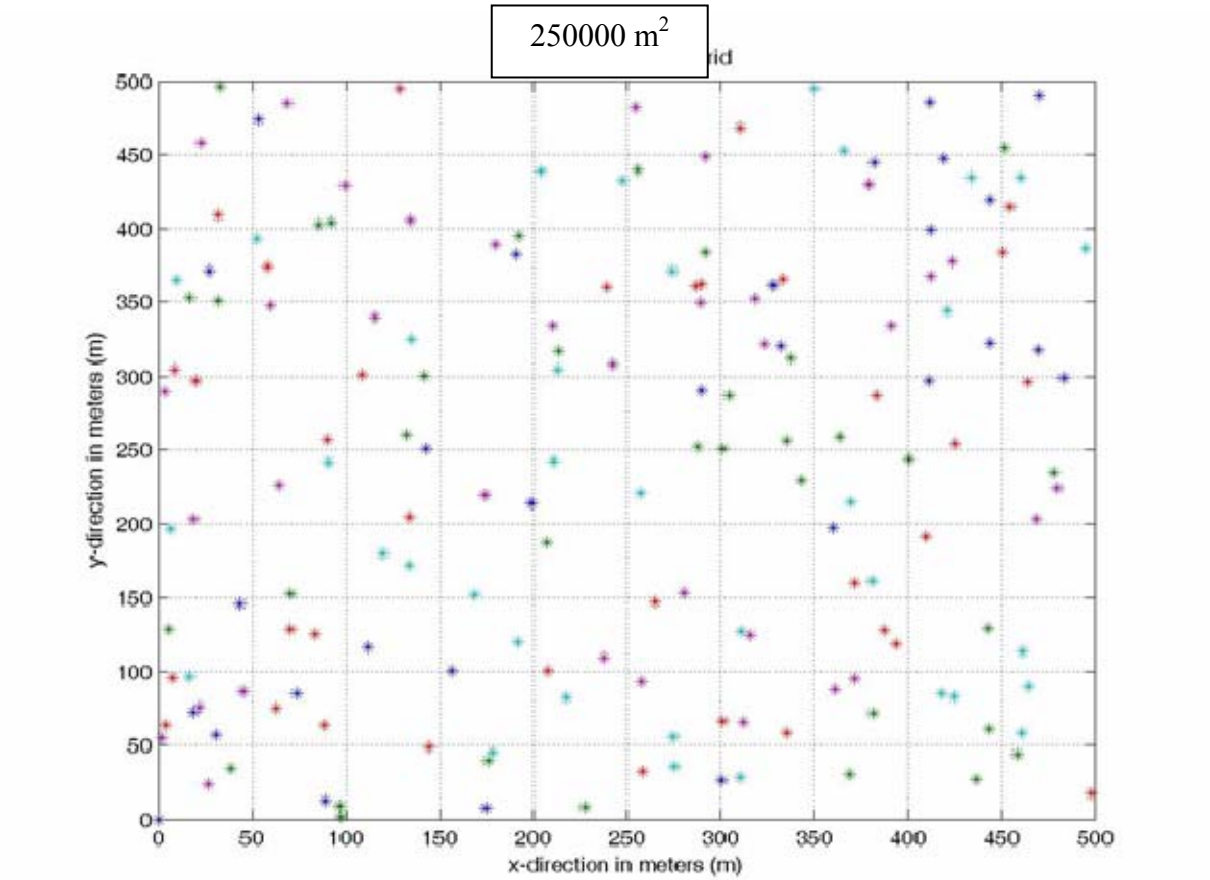


Figure 124. 175 randomly chosen sensors from a node array with a node density ≥ 0.0007 sensors/m^2.

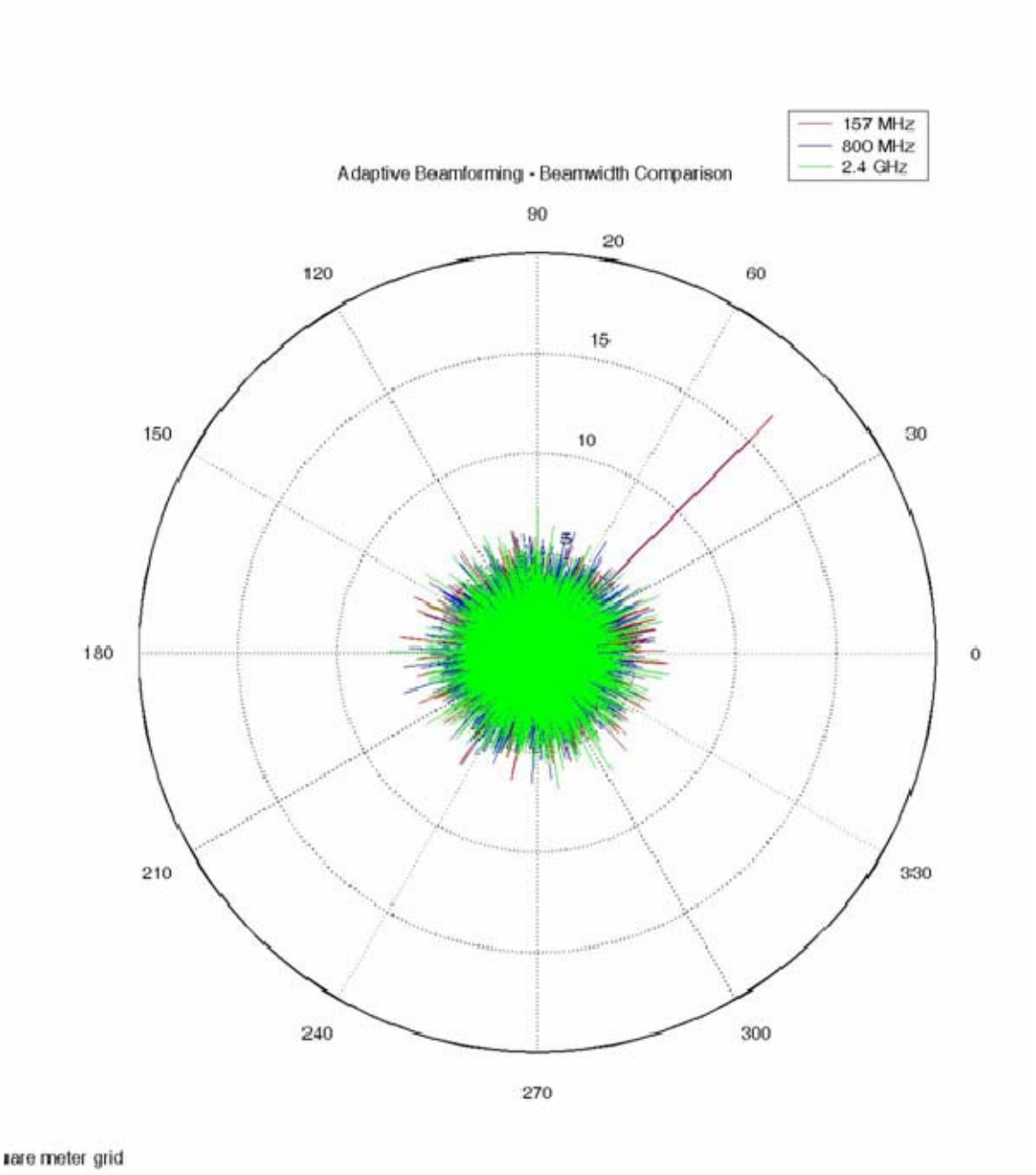


Figure 125. Array Pattern for a node density of ≥0.0007 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As expected the mainbeam beamwidth for all frequencies has continued to narrow.

Now an array size of $\Lambda \geq 250$ sensors will be investigated. The area, A_2, is altered between 25, 2500 and 2500000 m^2. This will have the affect of increasing the node densities at the respective areas.

Freq (MHz)	Noise Power (pW)	Area, A_2 (m^2)	Sample Size	# of iterations L_2	Number of Sensors per subset, P	Node Density (sensors/m^2)
157/800/2400	1	25	500	10	≥250	≥10
157/800/2400	1	2500	500	10	≥250	≥0.1
157/800/2400	1	250000	500	10	≥250	≥0.001

Table 25. Adaptive beamforming utilizing Method Two with a varying node.

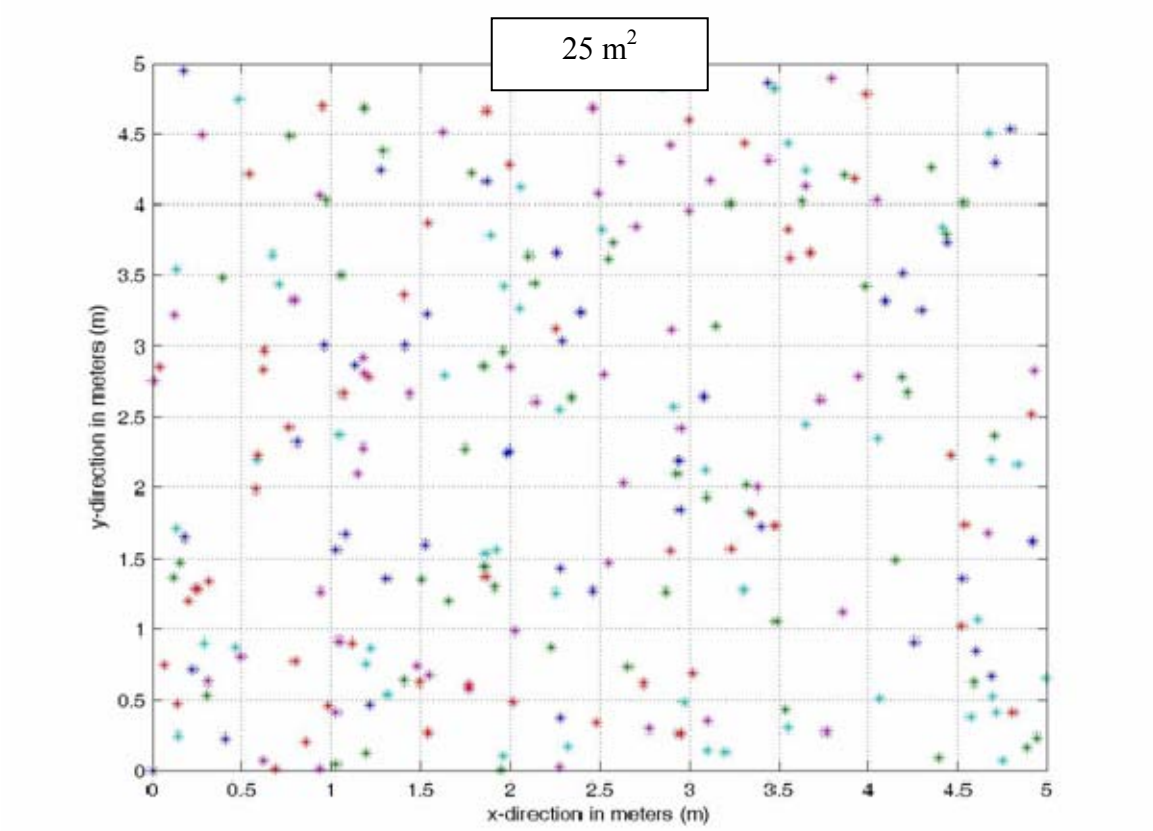


Figure 126. 250 randomly chosen sensors from a node array with a node density ≥ 10 sensors/m^2.

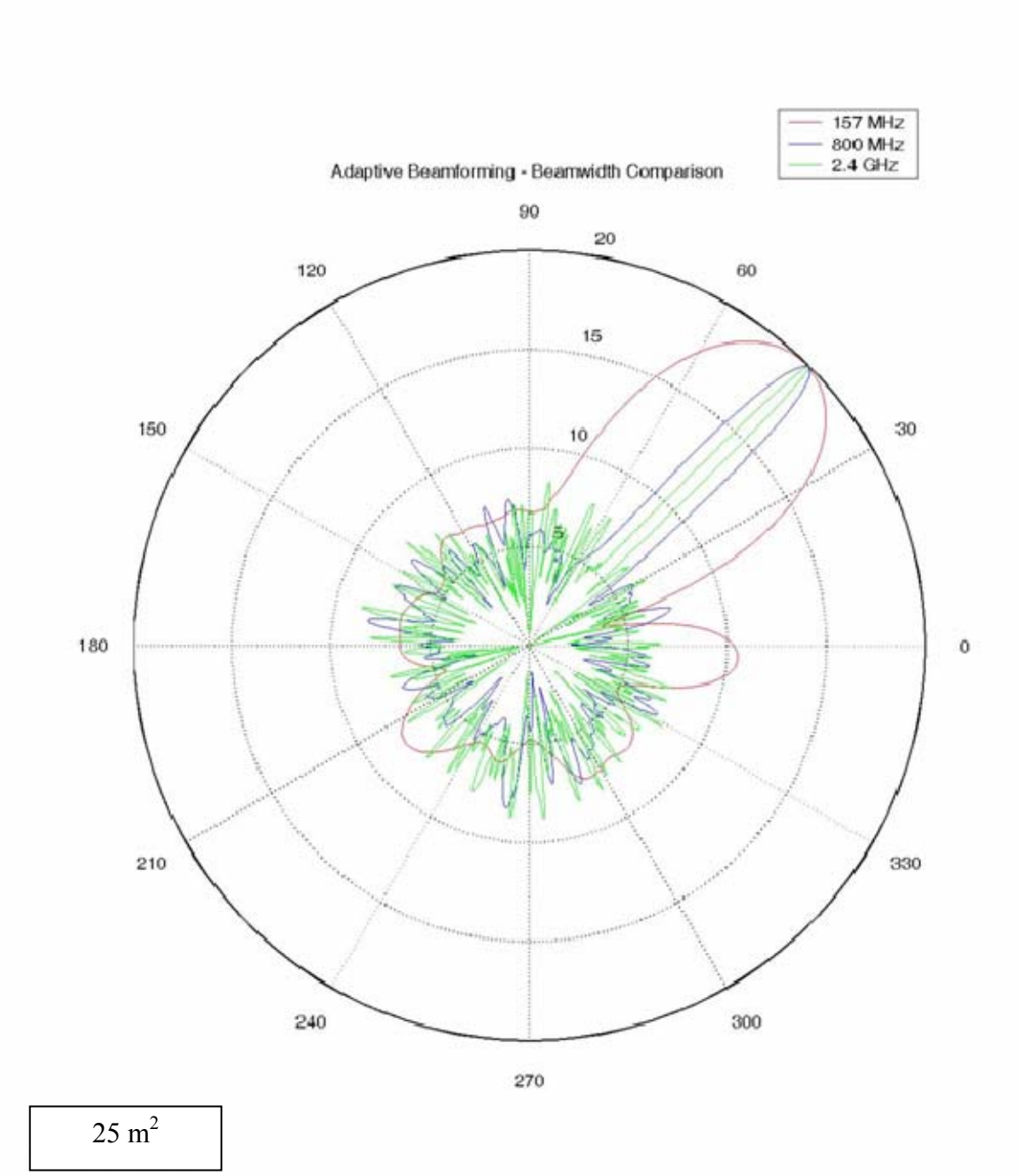


Figure 127. Array Pattern for a node density of ≥10 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

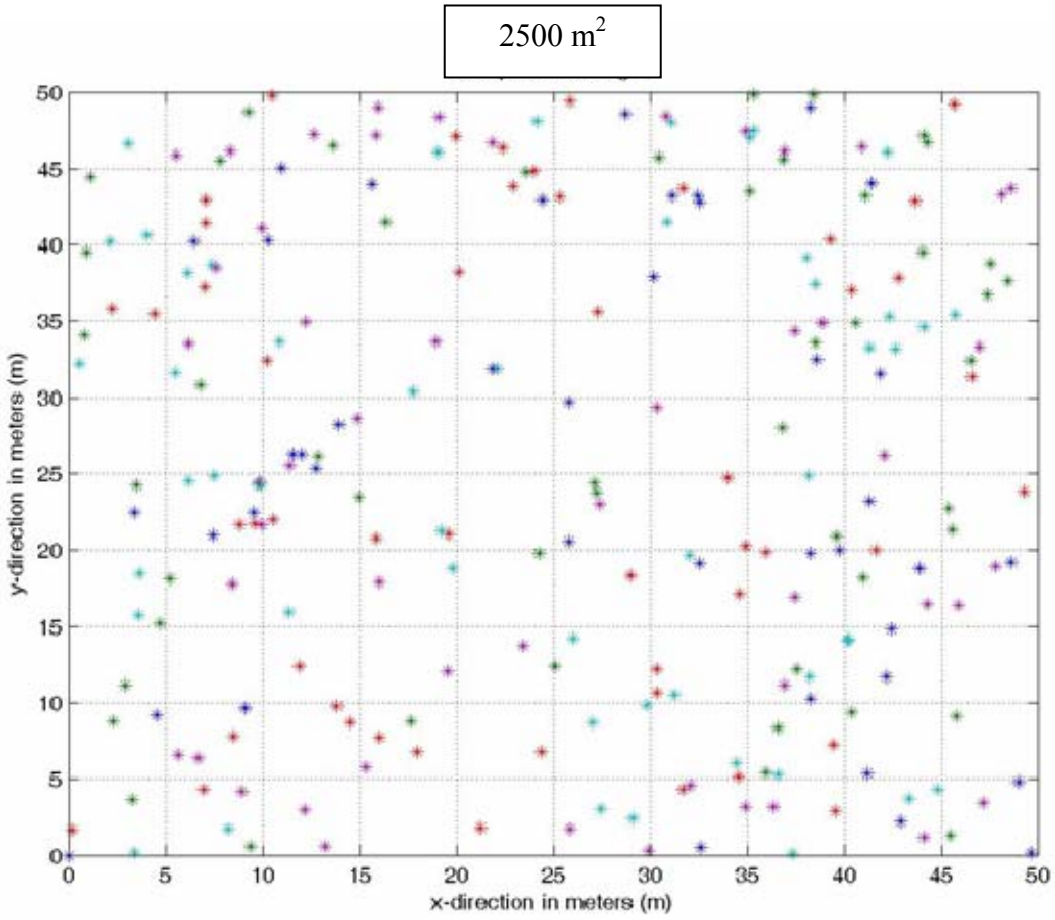
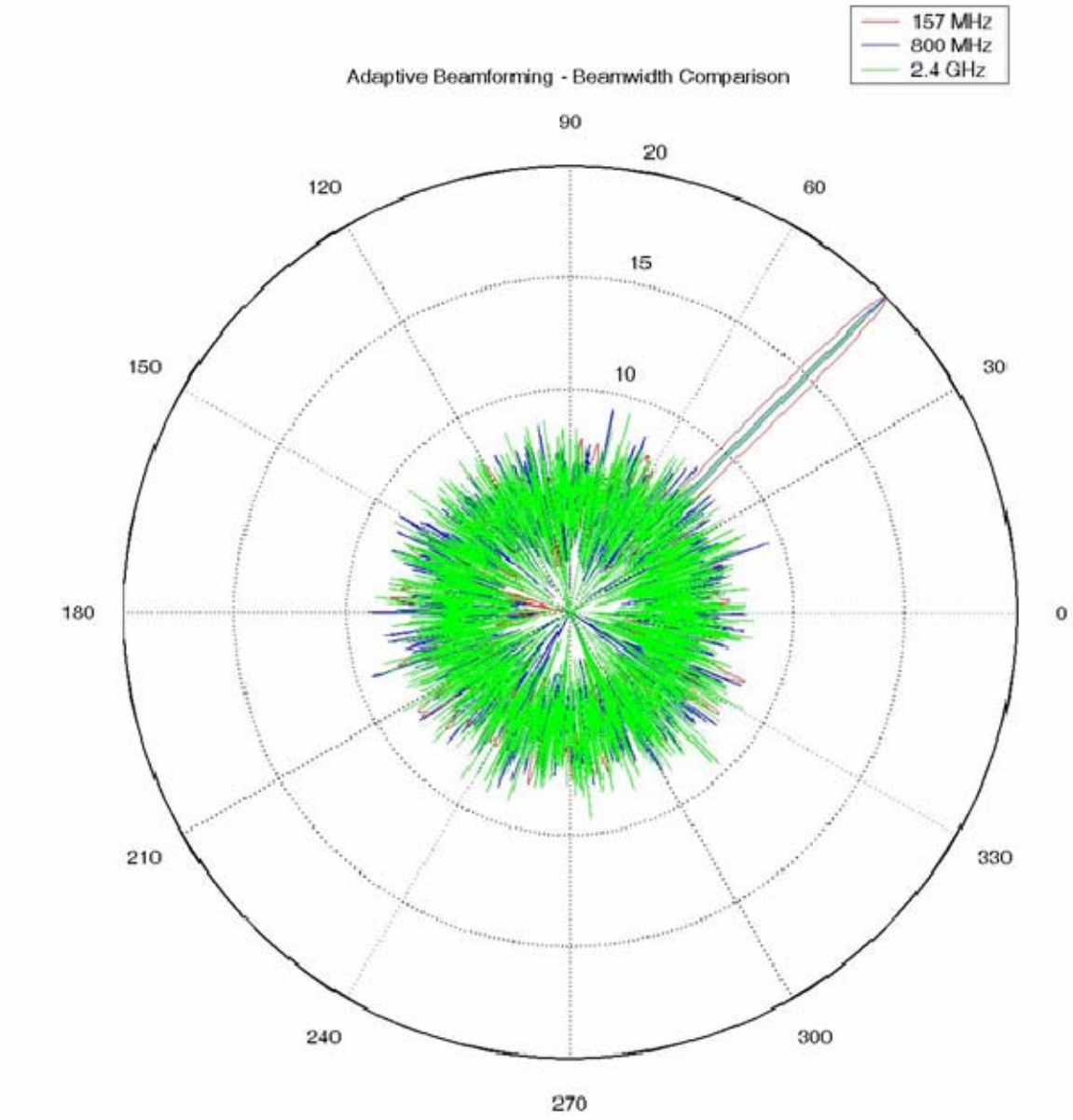


Figure 128. 250 randomly chosen sensors from a node array with a node density ≥ 1 sensors/m^2.



2500 m^2

Figure 129. Array Pattern for a node density of ≥1 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

As expected the mainbeam beamwidth has been narrowed considerably.

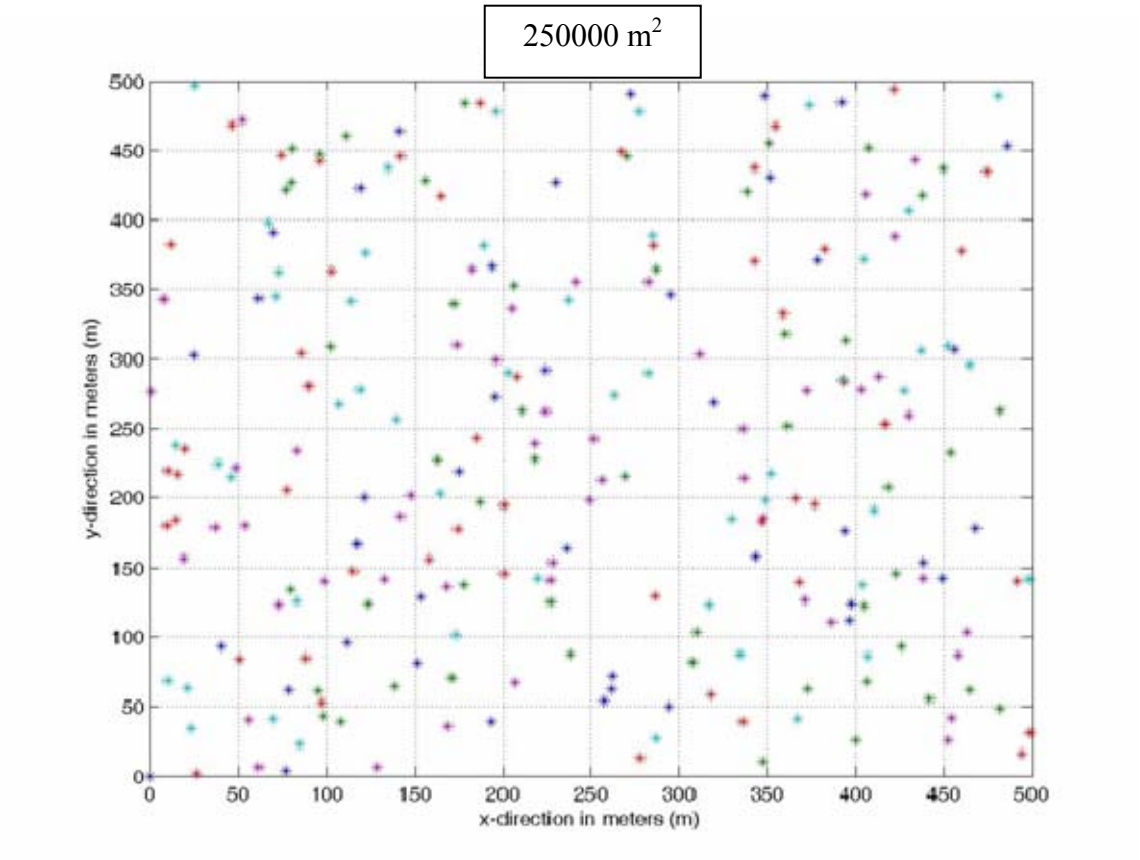


Figure 130. 250 randomly chosen sensors from a node array with a node density ≥ 0.001 sensors/m^2.

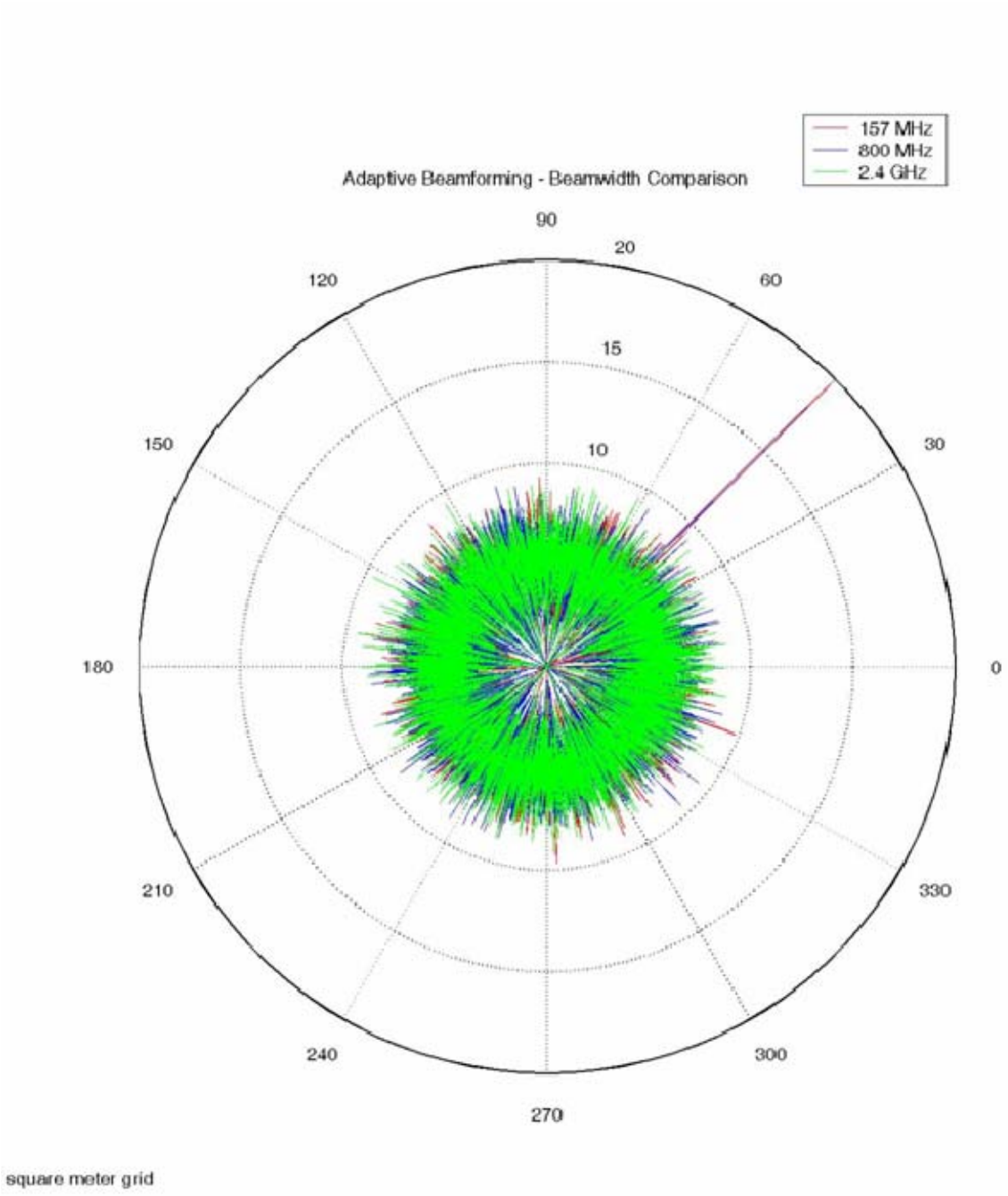


Figure 131. Pattern for a node density of ≥0.001 sensors/m^2 for frequencies 157 MHz (red), 800 MHz (blue) & 2400 MHz (green).

It has been shown that randomly selecting a different P_i subset of nodes per iteration from a large wireless antenna array with a suitable node density produces much better results. Now it will be shown how this method also decreases the computational burden on the central controller and the energy burden on the entire sensor grid. This method assumes that elements contained within the area, A_2, are only selected once, which is a reasonable assumption for a large array with equal probability of selection. The total transmissions required becomes

$$T_{\Psi} = \sum_{i=1}^{L_2} P_i = \Lambda, \tag{102}$$

where the subscript i in P_i is used to indicate that a different subset of nodes are chosen for each iteration. Again L_2 is use to represent Method Two. After modifying equation (100), for this methodology, which again assumes the worst case, that all nodes are a sleep and each node has to wait until the other sensors have transmitted, the total energy cost to form one beam in a given direction would be

$$\begin{aligned} \Gamma_{BF} &= \delta T_{\Psi} + \chi_1(\Lambda - 1) + \chi_2 \Lambda \\ &= \delta \sum_{i=1}^{L_2}(P_i) + (\Lambda - 1)(0.1\delta) + \Lambda(0.5\delta) \end{aligned}, \tag{103}$$

For this method, it is assumed that each node is only selected one time. This is a reasonable assumption if Λ is large and each node is equally likely to be selected. The energy cost is then distributed over all Λ sensors whereas in the first method the energy cost was spread over P sensors. This results in an energy cost per sensor of

$$\begin{aligned} \Xi &= \frac{\delta \sum_{i=1}^{L_2} P_i}{\sum_{i=1}^{L_2} P_i} + (0.1\delta) - \frac{0.1\delta}{\Lambda} + (0.5\delta) \\ \Xi &= \delta + (0.1\delta) - \frac{0.1\delta}{\Lambda} + (0.5\delta) \end{aligned}, \tag{104}$$

This is roughly a factor L_1 reduction in the energy cost per sensor compared to the first method. The savings is even more significant when realizing that $L_1 >> L_2$.

The data for this experiment is shown in Table 26. The same three frequencies were chosen for comparison. For comparison only the energy expended to actually transmit the signal is used. Two different subset areas were chosen to demonstrate the

effects on the beamwidth of the main lobe. In all cases $P_i = 25$ nodes and $L_2 = 10$ resulting in a total transmission cost of 250 transmissions. That is a reduction in total energy of 865δ joules per beam. Not only has the overall energy savings increased, but the energy savings per sensor are quite significant. For the previous example the energy cost per sensor using the first method would be 50.596δ joules/sensor, whereas for the second method the energy cost per sensor is δ joules/transmission, which is a significant energy savings.

Freq (MHz)	**# of Trans., T_Ψ**	**Spacing, A_2 (m^2)**	**Total Energy, Γ_Ψ (joules)**	**# of Iter., L_2**	**Energy per sensor, Ξ**
157 /800 /2400	250	25	399.9δ	10	1.6δ
157 /800 /2400	250	2500	399.9δ	10	1.6δ

Table 26. Total number of transmissions and energy expended utilizing Method Two.

As can be seen in the following figures, the mainbeam beamwidth has continued to get narrower as the central controller chose nodes that are further away from its own location. As expected as the main beam gain became narrower the side lobes increased. However, the main beam is large enough to mitigate the increase in side lobe level.

The simulations clearly support what can be analyzed in Equations (24) – (27), that beamforming process is highly dependent on the node density and the frequency of operations.

The results are as expected. Since the magnitude, after each iteration is added to the previous result and each iteration is allowing a new P number subset perspective the overall result is much better. The initial method while requiring less real sensors would be unable to perform the necessary SIGINT functions.

3. Beamwidth

When conducting direction finding it is important to understand how to control the beamwidth of the beam. When in a search mode a wider beam would be preferred. After the signal has been localized it would be preferable to narrow the beam in order to enhance collection and/or facilitate offensive information operations. The approximations for beamwidth given earlier in Equations (24) – (27) are repeated here for convenience and shown as Equations (105) – (108)

$$\Theta_{elevation} = \sqrt{\left(\frac{1}{\cos^2(\theta_a)\left[\theta_{x_a}^{-2}\cos^2(\phi_0) + \theta_{y_a}^{-2}\sin^2(\phi_0)\right]} \right)}, \tag{105}$$

$$\Theta_{azimuth} = \sqrt{\left(\frac{1}{\left[\theta_{x_a}^{-2}\sin^2(\phi_0) + \theta_{y_a}^{-2}\cos^2(\phi_0)\right]} \right)}, \tag{106}$$

$$\begin{aligned} \theta_{x_a} &= \cos^{-1}\left[\cos\theta_a - 0.443\left(\frac{\lambda}{(Length + d)} \right) \right], \\ \theta_{y_a} &= \cos^{-1}\left[\cos\theta_a + 0.443\left(\frac{\lambda}{(Length + d)} \right) \right] \end{aligned}, \tag{107}$$

where $d=Length/N$ and N is the number of nodes. [3]

$$\begin{aligned} \phi_{x_0} &= \cos^{-1}\left[\cos\phi_0 - 0.443\left(\frac{\lambda}{(Length + d)} \right) \right], \\ \phi_{y_0} &= \cos^{-1}\left[\cos\phi_0 + 0.443\left(\frac{\lambda}{(Length + d)} \right) \right] \end{aligned}, \tag{108}$$

The solid angle and the directivity from Equation (28) and (29) are repeated for convenience and become Equations (109) and (110).

$$\Omega_A = \Theta_{elevation}\Theta_{azimuth}, \tag{109}$$

$$D \approx \frac{32400}{\Omega_A(\text{degrees})}, \tag{110}$$

As stated earlier an inverse relationship exists between the node density and the beamwidth. Likewise an inverse relationship can be observed between beamwidth and directivity. Specifically, given a network with K number of nodes dispersed over an area, A_1, the node density can be increased by selecting nodes that are dispersed within an area, A_2 where $A_2 \leq A_1$ to create the beam. Based on the size of the area, A_2, and the number of nodes, the beamwidth of the beam can be tailored to specific applications, specifically a wide beam for detection and a narrow beam for localization and/or collection.

Figure 132 illustrates that when the number of sensors is increased, but the length of the array is held constant, i.e., the node density increases, the beamwidth increases resulting in a loss of gain, which can be seen in Equations (105) and (108). In general the more sensors that are added the directivity, D, i.e., gain, G, in a particular direction increases, but it is assumed that as more sensors are added the area is increased due to the increase in the physical area of the antenna.

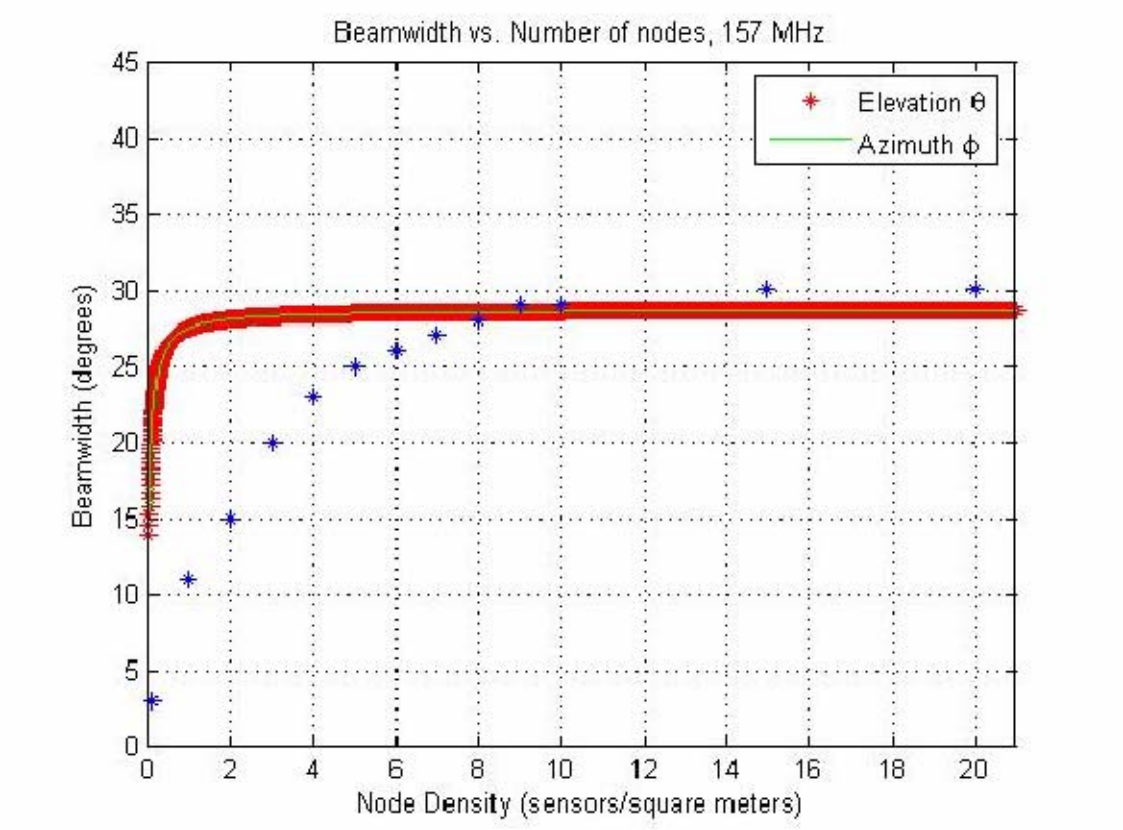


Figure 132. Beamwidth vs. Node Density where the blue asterisk represents measured values. Frequency is 157 MHz.

Figure 132 demonstrates and is consistent with Equations (105) – (108) that a limit is reached on how much control the central controller will have based on node densities and array distribution.

Nonetheless, this is an interesting discovery showing how the central controller can tailor the beamwidths to specific functions, e.g., search, localization, collection, offensive information operations, etc. When looking closely at the node densities represented in the above Figure 132 it becomes obvious that a limit on node densities has to be considered. Nodes spaced too close together would potentially create a mutual coupling problem that would need to be mitigated. More importantly, since it is envisioned that these sensors will need to be inconspicuous and randomly deployed, too many sensors in a given area could lead to a compromise of intent.

Now for a specific example, the software from Appendix I was modified to show the impact of sensor spacing and scanning a 360 degree arc. This modified software can be found in Appendix M. As seen in Tables 27 – 28 below a randomly distributed array containing $K = 1000$ sensors spread over an area of $A_1 = 100000\ m^2$ formed the basis of our sensor grid. The frequency was kept constant at 157 MHz.

Both examples will utilize the technique discussed in Chapter VI, Section B, subsection 2 to adaptively form the beams. The first example used sensors that were space within an area of $A_2 = 400\ m^2$ that contained the central controller. A randomly chosen subset of $P = 25$ nodes was created for each iteration, L_2 to form the beam. For this example $L_2 = 10$. Focusing on the 60° and 45° beams from Figure 133, it can be observed that a 360° scan could be conducted at 15° increments with sufficient overlap to not leave gaps in coverage, but provide enough spread to limit the total number of beams that needed to be formed. Thus, this 360° scan can be accomplished with 24 beams. Increasing the increments would decrease the number of beams to be created but also leaves the potential for creating gaps in coverage. Decreasing the gaps provides more overlap, but it also requires creating more beams.

Elevation Angle θ (radians)	Azimuth Angle φ (radians)	Number of sensors per subset, P	Area, A_2 (m^2)
π/4	π/3	25	400
π/4	π/4	25	400
π/4	π/6	25	400
π/4	π/7	25	400
π/4	π/8	25	400
π/4	π/9	25	400
π/4	π/10	25	400

Table 27. Adaptive Beamscanning utilizing Method Two where, A_2 = 400 m^2

Elevation Angle θ (radians)	Azimuth Angle φ (radians)	Number of sensors per subset, P	Area, A_2 (m^2)
π/4	π/3	25	40000
π/4	π/4	25	40000
π/4	π/6	25	40000
π/4	π/7	25	40000
π/4	π/8	25	40000
π/4	π/9	25	40000
π/4	π/10	25	40000

Table 28. Adaptive Beamscanning utilizing Method Two where, A_2 = 40000 m^2

The second example used sensors that were spaced within an area of $A_2 = 40000\ m^2$ that encompassed the central controller. Now investigating the use of sensors that are located further away from the central controller it can be seen that the beam has narrowed considerably. In Figure 134 below it can be observed that a reasonable scan increment would be 2°/beam in order to provide a sufficient overlap. This would require 180 separate solutions to be computed.

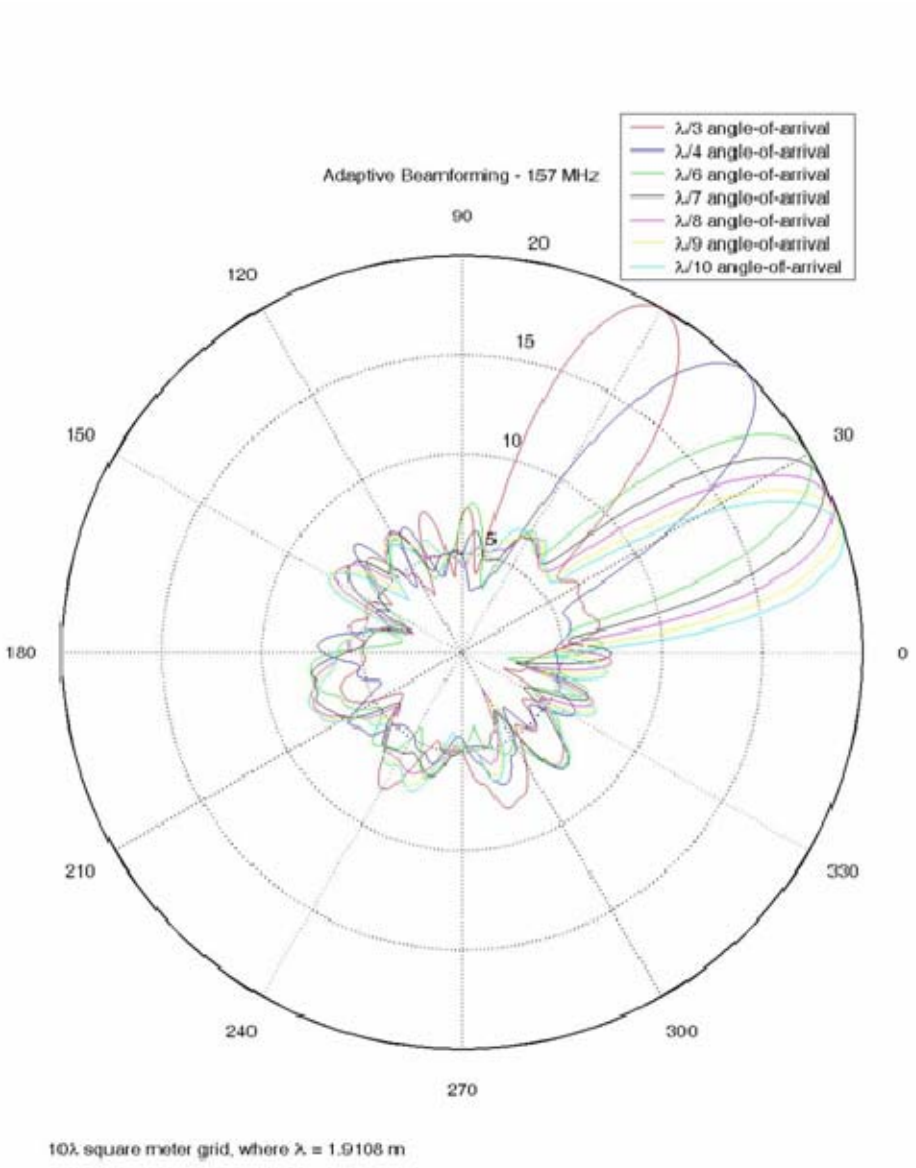


Figure 133. Array pattern for a beamscan at 400 m^2 with ≥ 250 nodes utilizing Method Two.

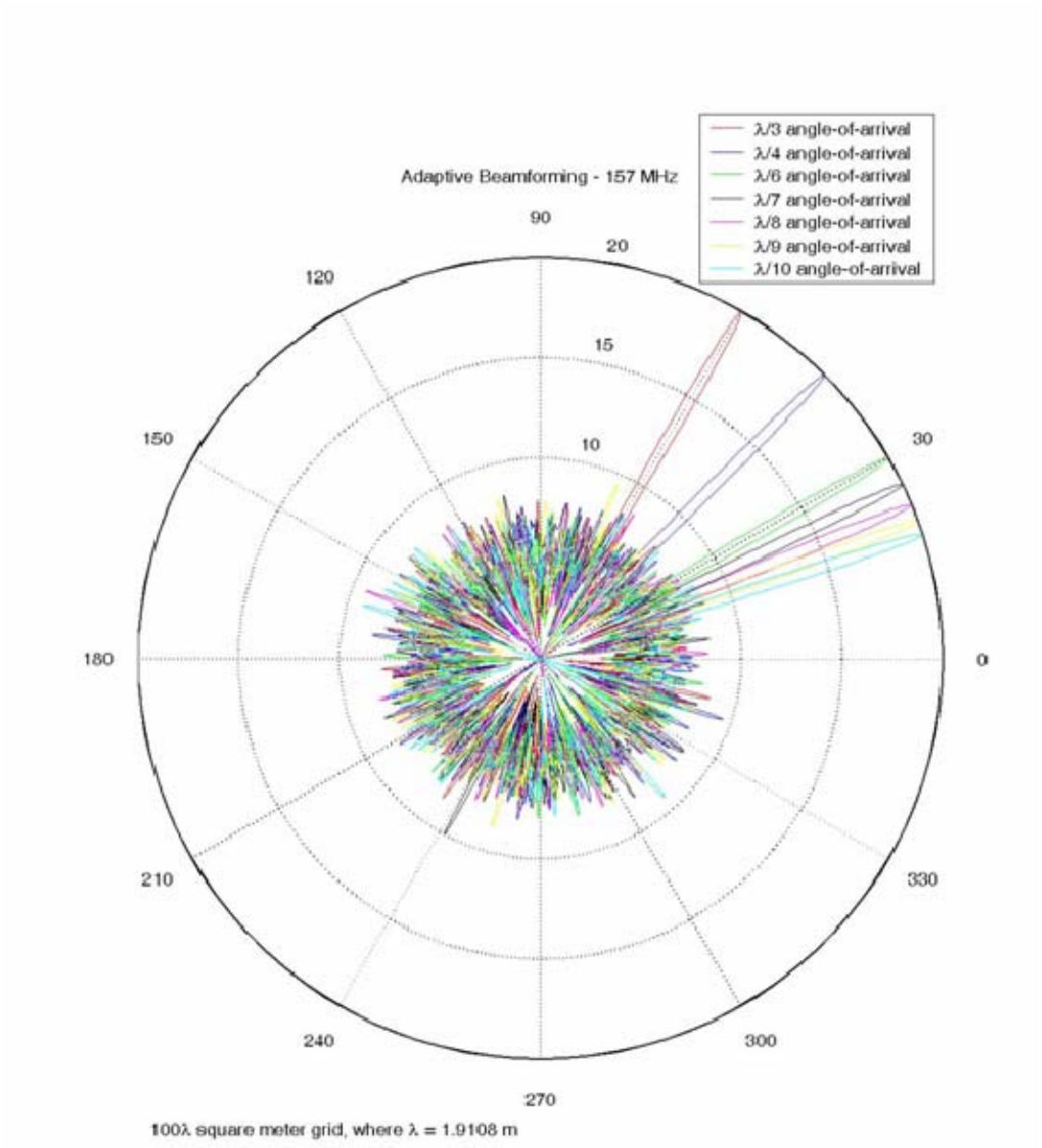


Figure 134. Array pattern for a beamscan at 40000 m^2 with ≥ 250 nodes utilizing Method Two.

The above results are consistent with the previous analysis that the frequency of the SOI, and node density have the greatest effect on beamwidth. The central controller

can vary the node density by selecting larger areas to create the subsets. The subset creation will be used to transition the beam from a signal search mode to signal prosecution mode.

This section has demonstrated how to control the beamwidth of the mainbeam. Also, shown was how a limit exists on how wide a beam can become given a certain node density. This limit far exceeds the practical limit of the size of the sensor. Therefore, this research will use the array pattern displayed in Figure 133 for search. The array pattern displayed in Figure 134 will be used for a more focused sustained collection and/or offensive information operations posture. It is envisioned that both sensor grids used to calculate the beams will be subsets of a larger grid being controlled by the same beam controller.

This chapter has provided the reader an insight into time difference of arrival and beamforming. The next chapter will provide a more indepth analysis into the speed and energy tradeoffs between time difference of arrival and beamforming

VII. DIRECTION FINDING METHODOLOGY

The proposed direction finding methodology for a SIGINT/IW sensor grid using a distributed antenna array is described as follows. The SIGINT/IW wireless sensor network will consist of a large number of small sensor nodes that are densely deployed over an area of interest to acquire information about signals of interest. These sensor nodes collaborate among themselves to form an ad-hoc network and disseminate the collected target information to a centralized gateway node, i.e. UAV, UUV, etc. The objective is to give national decision makers or combatant commanders the capability to have a relatively inexpensive solution for sustained SIGINT/IW or a rapidly deployable SIGINT/IW alternative.

A two-tiered hierarchical clustering sensor network architecture is assumed. Distributed array establishment and coordination will be achieved by using clustering algorithms to form a hierarchical architecture. In this research it is assumed that the clustering functions to create a sensor grid have been completed. It is assumed that perfect node location has been determined and that this data is available to the central controller. A central controller (i.e. UAV, UUV, etc.) is then tasked with maintaining frequency, phase and data synchronization among the remaining nodes (or secondary nodes) within the cluster. The primary node acts as a reference antenna by gaining initial intercept and frequency-of-interest (FOI) determination. The primary node will then coordinate the DOA determination. Once the DOA of the SOI has been determined the primary node will coordinate the beamforming process to direct an antenna beam towards the SOI in an adaptive manner in order to maximize collection or facilitate offensive information operations.

Now a more detailed discussion will be offered into how best to conduct direction finding with the unique architecture of the wireless SIGINT/IW antenna network. This researched focused on two competing methodologies for direction-finding within a wireless antenna array network with the specific focus being on rapid resolution of a line-of-bearing. The first methodology is adaptive beamforming that performs a beamscan similar to a phased-array radar. The second methodology uses a combination of TDOA

and adaptive beamforming. The TDOA lines-of-bearing are calculated using a combination of the Cross-Ambiguity Function to calculate the actual time difference and the Newton-Raphson technique to compute a geolocation. In both cases the adaptive beams are formed using a LMS algorithm. In this chapter, these methods have been adapted to the unique random distribution of the wireless sensors. This comparison is done to find the most expeditious way to calculate a solution and determine the angle-of-arrival.

A. ENHANCED COLLECTION METHODOLOGY (ECM)-1

Now a look at the time required to perform a combination TDOA and beamscanning solution. The same set-up requirements from Section A are assumed.

Step 1. The central controller acts as a reference antenna by gaining initial intercept and FOI determination.

Step 2. Based on the analysis in Chapter VI, Section A, subsection 2 the central controller will then determine the TDOA collected from the Ω sensors furthest from its location. Each sensor forms a two-node collector pair with the central controller. The central controller will utilize the cross-ambiguity function presented earlier to calculate the TDOAs between each of these Ω sensors and the central controller.

Step 3. The central controller will then use the modified Newton-Raphson technique to localize the signal-of-interest to a specified quadrant.

Step 4. The central controller will then coordinate the DOA determination using beamscanning. The central controller will create a subset of Λ elements randomly spread over an area, A_2, where $\Lambda \leq K$ and $A_2 \leq A_1$. The central controller will then select P_i nodes contained within an area, A_2, where $P << \Lambda$. Following the algorithm in Chapter VI, Section B, subsection 2, the central controller will create L subsets utilizing a different P_i nodes for each iteration to create the appropriate beam for a given (θ_a, ϕ_0).

Step 5. As stated above, in Section A, based on the results of this search then the beam controller can randomly choose sensors that are dispersed within a larger area to

further narrow the beam into the target direction in order to maximize collection or facilitate offensive information operations.

Now the total time required to do combination TDOA and beamforming can be defined as follows:

$$t_{TDOA/\Psi} = t_{TDOA} + t_{BF}\,, \tag{111}$$

where t_{TDOA} is the time to required to localize the target and t_{BF} is the time required to form one beam. After substituting Equation (78) into Equation (111) and combining like terms the total time required can be rewritten as

$$t_{TDOA/\Psi} = \Omega\varsigma\Upsilon + \Omega\varepsilon + \Omega t_1 + \Omega t_2 + t_\Psi + \Lambda t_1 + \Lambda t_2\,, \tag{112}$$

where t_1 is the time required to transmit and t_2 is the time required to bring a sensor out of sleep mode. Equation (112) assumes the worst case that all the nodes need to be brought out of a sleep mode. In addition, it assumes that the nodes used in the TDOA calculations are not used in the beamforming calculations.

The total energy to enhance collection using a combination of TDOA and beamforming is defined as

$$\Gamma_{TDOA/\Psi} = \Gamma_{TDOA} + \Gamma_{BF}\,, \tag{113}$$

where again Γ_{TDOA} is defined in Equation (76) as the energy required for TDOA and Γ_{BF} is defined in Equation (103) as the energy required to form one beam using the multiple random selections method. After substitution, Equation (113) can be rewritten as follows:

$$\begin{aligned}\Gamma_{TDOA/\Psi} &= \Omega\delta\Upsilon + (\Omega-1)\Upsilon\chi_1 + \Omega\chi_2 + \delta\sum_{i=1}^{L_2} P_i + (\Lambda-1)\chi_1 + \Lambda\chi_2 \\ \Gamma_{TDOA/\Psi} &= \Omega\delta\Upsilon + (\Omega-1)\Upsilon(0.1\delta) + \Omega(0.5\delta) + \delta\sum_{i=1}^{L_2} P_i + (\Lambda-1)(0.1\delta) + \Lambda(0.5\delta)\end{aligned}, \tag{114}$$

B. ENHANCED COLLECTION METHODOLOGY (ECM)-2

The proposed approach for a wireless antenna sensor grid using beamscanning only is described as follows. For our sensor grid, the central controller selects the number

of array elements needed to form the desired beam. The central controller in effect creates an array, which is merely a subset of the total array. For example, a SIGINT grid that contains K elements randomly spread over an area, A_1.

Step 1. The central controller acts as a reference antenna by gaining initial intercept and FOI determination.

Step 2. The central controller will then coordinate the DOA determination using beamscanning. The central controller will create a subset of Λ elements randomly spread over an area, A_2, where $\Lambda \leq K$ and $A_2 \leq A_1$. The central controller will then select P_i nodes contained within an area, A_2, where $P << \Lambda$. Following the algorithm in Chapter VI, Section B, subsection 2, the central controller will create L subsets utilizing a different P_i nodes for each iteration to create the appropriate beam for a given (θ_a, ϕ_0).

Step 3. The beam controller will repeat Step 2 until a 360 degree scan has been completed.

Step 4. Once the SOI has been localized to a specific quadrant then the central controller will repeat step 2, except the central controller will use nodes that are dispersed over a larger area, A_3 to create the P_i node subset. This will allow the central controller to create a narrower beam to localize the SOI to a specific sector of the quadrant. More sensors can be incorporated into the adaptive beamforming solution if higher gain is required.

Step 5. Based on the results of this search the beam controller can randomly incorporate sensors that are dispersed within an even larger area, A_4, to further narrow the beam into the target direction in order to maximize collection or facilitate offensive information operations.

Now the total time required to do beamscanning only can be defined as follows:

$$t_{bs} = \eta t_{\Psi} + \eta \Lambda t_1 + \Lambda t_2, \tag{115}$$

where t_{Ψ} is the time to required to form one beam, as defined earlier, and η is the number of beams required to conduct a 360° scan. η is a function of the solid angle of the beam and the user defined overlap desired. t_1 is the time to transmit and t_2 is the time required

to bring the node out of a sleep cycle. Equation (115) assumes the worst case therefore providing an upper bound on the time required to perform ECM-2. Equation (115) assumes that a transmitting node will have to wait until all the other nodes are finished transmitting. Likewise Equation (115) assumes that all nodes participating in ECM-2 will only need to be brought out of a sleep mode one time.

The total energy to enhance collection using beamscanning only is defined as

$$\begin{aligned}\Gamma_{bs} &= \eta\left(\delta T_{\Psi}\right)+\eta\left(\Lambda-1\right)\chi_1+\Lambda\chi_2 \\ &= \eta\left(\delta T_{\Psi}\right)+\eta\left(\Lambda-1\right)\left(0.1\delta\right)+\Lambda\left(0.5\delta\right)\end{aligned}, \tag{116}$$

where η is the number of beams that need to be formed. Equation (116) is similar to Equation (103) except that multiple beams need to be formed. After substitution, Equation (116) can be rewritten as follows:

$$\Gamma_{bs} = \eta\left(\delta\sum_{i=1}^{L_2} P_i\right)+\eta\left(\Lambda-1\right)\left(0.1\delta\right)+\Lambda\left(0.5\delta\right), \tag{117}$$

Equation (117) assumes the worst case scenario in order to provide an upper bound on energy usage. Equation (117) assumes that energy is spent waiting to transmit and that each node had to be brought out of a sleep cycle. The first term in Equation (117) accounts for the number of beams to be formed. The second term in Equation (117) accounts for the energy expended waiting to transmit. Finally, the third term in Equation (117) is the energy expended to bring the participating nodes out of a sleep mode. It is assumed that the nodes will only need to be brought out of a sleep mode one time.

Now a look at the combination of TDOA and beamforming.

C. COMPARISON

The energy cost using the combination of TDOA and beamforming will always be less than the beamscanning only method as long as the following criteria is met:

$$
\begin{aligned}
&\Gamma_{TDOA/\Psi} \le \Gamma_{bs} \\
&\Omega\delta\Upsilon + (\Omega - 1)\Upsilon(0.1\delta) + \Omega(0.5\delta) + \delta\left(\sum_{i=1}^{L_2} P_i\right) + (\Lambda - 1)(0.1\delta) + \Lambda(0.5\delta) \\
&\le \eta\delta\left(\sum_{i=1}^{L_2} P_i\right) + \eta(\Lambda - 1)(0.1\delta) + \Lambda(0.5\delta) \\
&\Omega\Upsilon + \Omega\Upsilon(0.1\delta) - \Upsilon(0.1\delta) + \Omega(0.5\delta) \le (\eta - 1)\sum_{i=1}^{L_2} P_i + (\Lambda - 1)(\eta - 1)(0.1\delta) \\
&T_{TDOA} + \Omega\Upsilon(0.1\delta) - \Upsilon(0.1\delta) + \Omega(0.5\delta) \le (\eta - 1)T_\Psi + (\Lambda - 1)(\eta - 1)(0.1\delta)
\end{aligned}
\quad , (118)
$$

Upon close inspection of Equation (118), it can clearly be seen that ECM-1 will always use less energy than ECM-2 as long as the number of transmissions to perform the TDOA plus the additional factor associated with the energy spent waiting to transmit and the energy required to bring a sensor out of a sleep cycle is less than the number of transmissions required to conduct the 360° beamscan minus the target beam. The target beam will be formed in both cases.

Additionally, the combination of TDOA and beamforming will always be faster as long as the following criteria is met:

$$
\begin{aligned}
&t_{TDOA/\Psi} \le t_{bs} \\
&\Omega\varsigma\Upsilon + \Omega\varepsilon + \Omega\Upsilon t_1 + \Omega t_2 + t_\Psi + \Lambda t_1 + \Lambda t_2 \le \eta t_\Psi + \eta\Lambda t_1 + \Lambda t_2 \\
&\Omega\varsigma\Upsilon + \Omega\varepsilon + \Omega\Upsilon t_1 + \Omega t_2 \le (\eta - 1)t_\Psi + \Lambda(\eta - 1)t_1 \\
&t_{TDOA} + \Omega\Upsilon t_1 + \Omega t_2 \le (\eta - 1)t_\Psi + \Lambda(\eta - 1)t_1
\end{aligned}
\quad , (119)
$$

Equation (119) shows that ECM-1 will be faster than ECM-2 will be faster as long as the time required to form the beams, less the target beam, plus the amount of time spent waiting to transmit and the time required to bring a sensor out of a sleep mode is greater than the time to determine the target line-of-bearing the combination of TDOA and beamforming will always be faster than beamscanning only.

Focusing on the 60° and 45° beams from Figure 133, it can be observed that a 360° scan could be conducted at 15° increments with sufficient overlap to not leave gaps in coverage. This can be accomplished with $\eta = 24$ beams. Using the beamwidth from Figure 134 it would require 6850 time units to scan 360°.

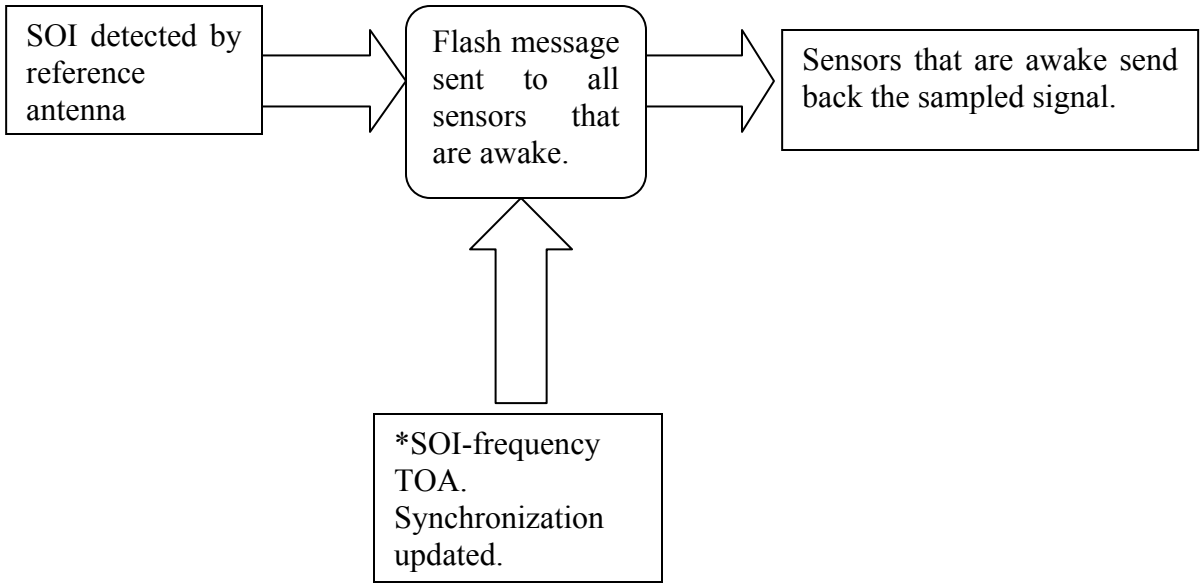
The energy and time cost are dependent on the number of beams that need to be formed for beamscanning only. As stated earlier, the number of beams required would be based on the solid angle of the beam and the desired overlap of the beams. Table 29 depicts the energy cost and times for various η. The `tic-toc` command in MATLAB® was used to determine that $t_{\psi} = 25$ time units were required to create a beam, ε was determined to be equal to 0.1 time units and ς was determined to be 2 time units per CAF estimate for each two-node collector pair. These times would vary based on hardware and software implementation. Additionally, it is expected that these times would be much lower in a real world implementation. From Table 26, the energy to form one beam is 399.9δ joules. From Equation (76), the total energy to do TDOA is 45δ joules, where $\Omega =$ 4 sensors.

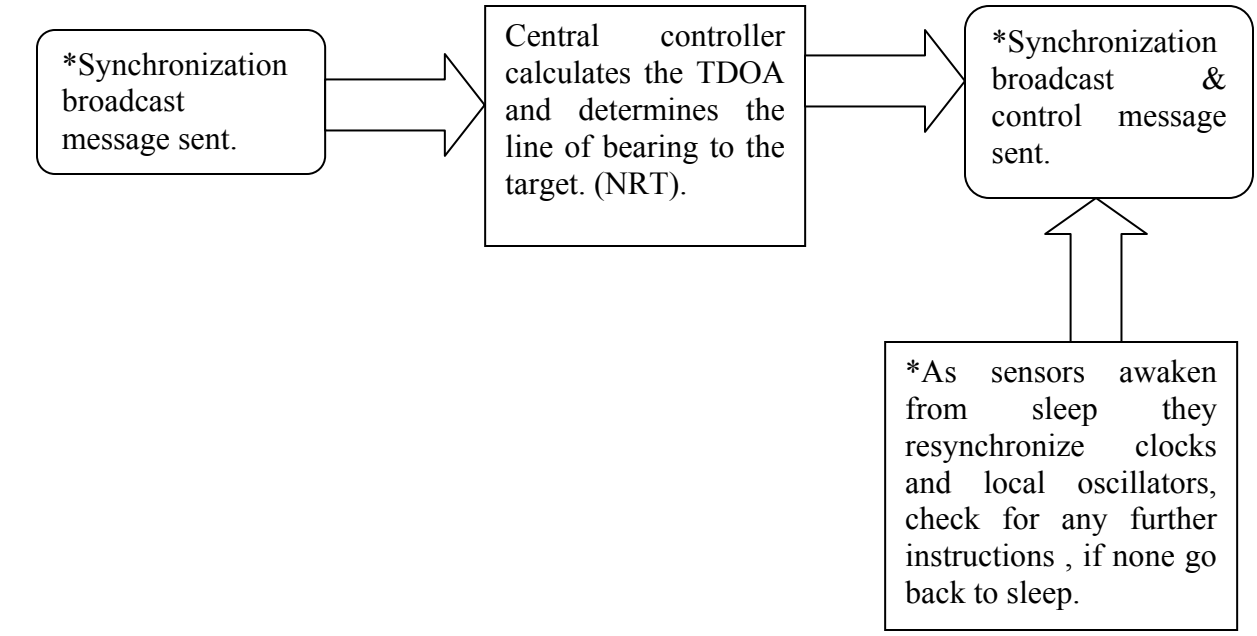
# of beams, η	**Γ_{bs}, (joules)**	**$\Gamma_{TDOA/\Psi}$, (joules)**	**t_{bs}, (s)**	**$t_{TDOA/\Psi}$, (s)**
10	2874δ	444.9δ	3000	612.4
24	6722.6δ	444.9δ	6850	612.4
30	8372δ	444.9δ	8500	612.4

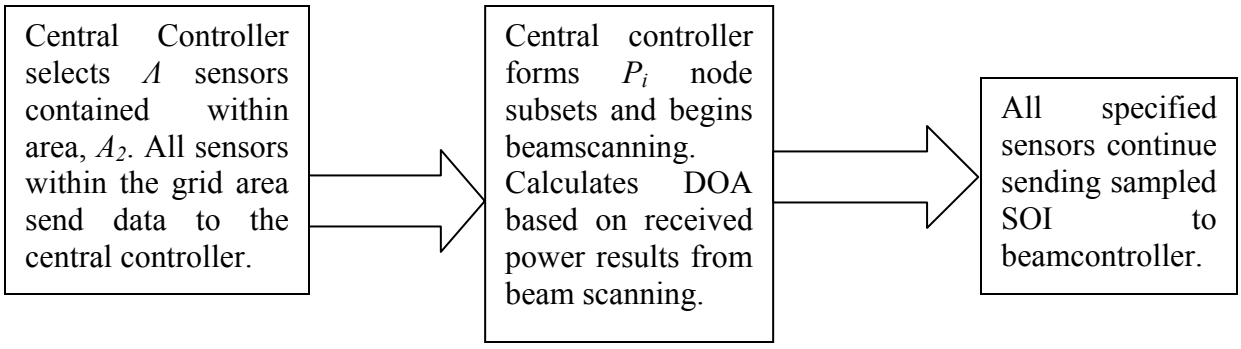
Table 29. Energy and Time Cost comparison between ECM-1 vs. ECM-2.

It can be seen from the results shown in Table 29 that the combination of TDOA and beamforming in most practical cases will always be faster and more energy efficient than the beamscanning only method.

Once the correct beam has been formed then the determination on whether to continue prosecution will need to be made. In addition, using the knowledge gained on how to affect the beamwidth of the beam, a narrower beam can be created for sustained collection and/or offensive information operations. Figure 135, shows a schematic diagram of the information flow in the SIGINT/IW wirelessly distributed antenna system.


SOI detected by reference antenna
Flash message sent to all sensors that are awake.
Sensors that are awake send back the sampled signal.
*SOI-frequency TOA. Synchronization updated.


*Synchronization broadcast message sent.
Central controller calculates the TDOA and determines the line of bearing to the target. (NRT).
*Synchronization broadcast & control message sent.
*As sensors awaken from sleep they resynchronize clocks and local oscillators, check for any further instructions , if none go back to sleep.


Central Controller selects Λ sensors contained within area, A_2. All sensors within the grid area send data to the central controller.
Central controller forms P_i node subsets and begins beamscanning. Calculates DOA based on received power results from beam scanning.
All specified sensors continue sending sampled SOI to beamcontroller.

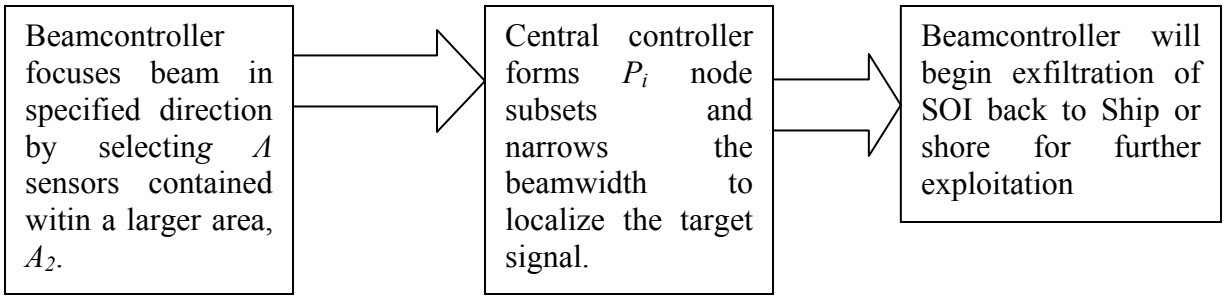


Figure 135. Schematic diagram of the information flow in a SIGINT/IW wirelessly distributed antenna system. *Note – Synchronization implies both timing and local oscillator synchronization. Initial Conditions: time, $t = 0$. The following assumptions apply: All participating sensors are on; All sensors are interconnected and synchronized (timing and local oscillator); All nodes have connectivity with the central controller node; Sensor Receivers monitor a specified band, i.e. AD8347 (800 MHz – 2.5 GHz).

In this chapter a methodology for rapid resolution to determine lines of bearing in a wireless antenna array network has been presented. Two competing methodologies were discussed. One method used beamscanning exclusively and the other method used a combination of TDOA and beamscanning. The metric for determining the best method was the time required to perform each method. It was shown that the combination TDOA and beamscanning method is far superior to the beamscanning only method.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. SUMMARY

In this research, a novel approach for conducting signals intelligence from a distributed network of wireless nodes was developed. The primary objective of this research is enhancing collection in a specified target direction. The focus was on enhancing RF signal collection through beamforming in a wireless antenna system. A variety of techniques were investigated with a combination of TDOA and adaptive beamforming showing the most promising results. The combination of the two techniques dubbed Enhanced Collection Methodology (ECM) – 1 provides rapid acquisition, localization, collection and/or the option of conducting offensive information operations, against a particular SOI.

A. CONCLUSIONS

In the development, of ECM-1 two specific areas of conducting RF collection were addressed in this research. One is the time required to determine the target direction and the time required to form the beams. The other is the energy consumption involved in developing these solutions. Two competing enhanced collection methodologies (ECM) were developed and researched, ECM-1 and ECM-2. ECM-1 uses a combination of time difference of arrival (TDOA) and adaptive beamforming. ECM-2 uses adaptive beamforming that performs a beamscan similar to phased-array radars.

Additionally, two competing methods for forming the beams were created, Method One and Method Two. Method One created a subset of sensors from a randomly distributed antenna array. This method uses data exclusively from these elements and increases the number of iterations until the desired beam is formed. Method Two creates a new subset of sensors, from a randomly distributed array, for each iteration, until the desired beam is formed.

Analytic expressions were derived that proved that Method Two provides better power management across the sensor network. Additionally, Method Two was able to mitigate the grating lobes that are present in arrays where the antenna element spacing is greater that half-wavelength.

Another important discover within this research was the ability of the central controller to control the beamwidth, therefore the gain of the main beam. By randomly selecting nodes that were distributed over different areas the central controller, in effect manipulates the node density. Outside of the frequency of the tartget signal the node density was shown to have the greatest affect on the beamwidth of the main beam. The significance of this discovery is that no additional energy burden was placed on the network.

Analytic expressions, for energy consumption and time to determine a line of bearing solution, were derived to compare ECM-1 to ECM-2. The energy consumption for ECM-1 is based on the energy required to determine the line of bearing to the target using time difference of arrival plus the energy required to form the beam. This method takes advantage of the spatial diversity of the nodes. Multiple estimates taken from a select number of nodes located furthest from the central controller reduces the number of nodes required to determine the target direction. By reducing the number of participating nodes the overall energy consumption of the network is reduced.

The energy required for ECM-2 is based on the requirment to form multiple beams in order to provide sufficient coverage for a given area. Beamforming was shown to require a lot of energy. Thus having to form multiple beams placed a tremendous energy burden on the network and was shown to be too costly.

The time required to conduct ECM-1 is based on the time required to determine the target line of bearing through time difference of arrival plus the time required to form the beam in the target direction. This method also takes advantage of the spatial diversity of the individual nodes. Multiple estimates taken from a select number of nodes located furthest from the central controller is all that was required to determine a line of bearing to the target. After the line of bearing to the target has been determined then only the one beam needs to be formed.

In comparison, the time needed to conduct ECM-2 is based on the requirment to form multiple beams in order to search a given area. The number of beams required to search a given area is a user defined requirement that is based on the beamwidth of the

main beam. Beamforming was determined to be a time consuming evolution and the requirement to form multiple beams to determine the target direction proved too costly.

ECM-1 was shown to be far superior to ECM-2 in both energy consumption and the time required to formulate a solution.

B. AREAS OF FUTURE RESEARCH

A few potential follow-on research areas will now be discussed. One area of research is the ability of this methodology to prosecute mobile targets. Another research area is the ability of this network to work against wideband signals. ECM-1 should be robust enough after some minor modifications to address these new requirements.

Characterizing the types of traffic will be crucial. This dissertation assumed that sufficient buffer space was resident at the individual nodes. Understanding the type of traffic would be critical in selecting the appropriate medium access control scheme, as well as, buffer sizes at the individual nodes. If buffers overflow then critical data needed to perform the beamforming would be lost. Selecting the appropriate MAC will be critical in insuring that sensors can access the network and transmit their data reliably with a minimum of retransmissions before their buffers overflow. It is expected that the same data will be arriving at the nodes at slightly different time intervals, therefore a TDMA scheme would probably be the best option.

Other issues are the synchronization of the nodes. It was assumed in this dissertation that all sensor nodes were synchronized. It is believed that the synchronization demands will be such that the nodes will need to have constant communications with the central controller. If this is indeed the case then a separate channel will probably be needed to accommodate this requirement. Both the data and synchronization channels will need to be at a minimum low probability of intercept (LPI) channels to prevent compromising the intent of this network and exposing it to various attacks. It would be optimal to have the channels be low probability of detection (LPD) signals, as well.

Due to the constant transmissions expected to keep the sensors synchronized research into power consumption will be important. The synchronization methods

discussed in this research are power intensive and assume that the nodes are connected to a reliable energy source, i.e., a ship's power system. Transmissions are one of the highest energy burdens on a sensor. Therefore, innovative solutions will be needed for the type of modulation scheme designed for synchronization.

A more intense study of the sample size should be investigated. In this research, 500 samples of data were used for calculating a solution. Reducing the sample size would reduce the transmission length and the loading of the network.

The use of mobile nodes will also merit further study, as well. In this dissertation, all nodes were assumed to be static. Having static nodes required a large number of nodes. If mobile nodes are used then a smaller number of nodes may be required.

However, what will be the impact on the beamforming process of the samples being taken at different instances in time? This will no doubt have a deleterious effect on the formation of the beams, but the question becomes how much.

Initial investigations show promising results. The direction finding method is similar to the first method discussed in Chapter VI. The central controller will select a subset of mobile nodes, P that are contained within an area, A_2. These P nodes will form a subset of array elements that the central controller will use to calculate the array weights. The LMS algorithm adaptively produces weights for the formation of the beam. These weights are then used to calculate the magnitude and angle of the beam. Next the central controller obtains more data from the same subset of P nodes selected earlier, except unlike in the first method these nodes have moved. Therefore, the new data will be taken from a different location. The new weights are then calculated and used to calculate the magnitude and angle of the beam in the desired direction. This result is then added to the previous calculations and then stored. This iterative process is continued for L iterations. Equation (94) is rewritten dropping the *i* dependence, to reflect the fact that the same P nodes are being used. This process is shown in Equation (98). The only difference will be in the beamforming equation. Equation (97) will be modified to account for the *z* component of our sensors as shown in Equation (120).

$$\Psi(\theta,\phi) = e^{\left(j\frac{2\pi}{\lambda}*(x\sin(\theta)*\cos(\phi)+y\sin(\theta)\sin(\phi)+z\cos(\theta))\right)}, \quad (120)$$

where λ is the wavelength of the signal in meters; x, y, and z are the location of the individual antenna elements in meters; and θ is the elevation angle and ϕ is the azimuth angle. This magnitude is then plotted in dB against all possible values of azimuth and elevation. This process is repeated until a desired solution has been found. MATLAB® code developed in [73] was modified to conduct a few simulations. This modified code can be found in Appendix K.

Something very interesting was discovered. If the $\cos(\theta)$ term in the exponent was dropped the beamwidth of the main lobe can be increased, thereby increasing the detection region in a search mode.

This method may show the viability of using a few nodes and being able to gain some of the same benefits that a larger array would provide. Certainly, these nodes would be more expensive than their static counterparts. This method could be applied to any type of mobile sensor, e.g. UAV, UUV, etc.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Adams, R. T., "Beam tagging for control of adaptive transmitting arrays," *IEEE Transactions on Antennas and Propagation*, Vol. AP-12, pp. 224 – 227, March 1964.

[2] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., "A Survey on sensor networks," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102 – 114, 2002.

[3] Balanis, C. A., *Antenna Theory Analysis and Design*, 2nd edition, Wiley, N. Y., 1997.

[4] Barric, G., Mudumbai, R. and Madhow, U., "Distributed beamforming for information transfer in sensor networks," *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pp. 81 – 88, 2004.

[5] Bartlett, M. S., "Smoothing Periodograms from Time Series with Continuous Spectra". *Nature*, 161:686–687, 1948.

[6] Blogh, J. S. and Hanzo, L., *Third-Generation Systems and Intelligent Wireless Networking, Smart Antennas and Adaptive Modulation*, IEEE Press, John Wiley & Sons, N. Y., 2002.

[7] Breslin, D. F., Adaptive Antenna Arrays Applied to Position Location, Master's Thesis, Virgina Polytechnic Institute, Blacksburg, VA., 1997.

[8] Bulusu, N., Heidemann, J., and Estrin, D., "GPS-less low cost outdoor localization for very small devices," IEEE Personal Comm., vol. 5, no. 5, pp. 28–34, October 2000.

[9] Chong, C. and Kumar, S., "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247 – 1256, August 2003.

[10] Clark, V., "Sea Power 21," *Proceedings*, October 2002, http://www.chinfo.navy.mil/navpalib/cno/proceedings.html, last accessed March 2005.

[11] dos Coelho, J. M. S., *Underwater Acoustic Networks: Evaluation of the Impact of Media Access Control on Latency, in a Delay Constrained Network*, Master's Thesis, Naval Postgraduate School, Monterey, CA., March 2005.

[12] Conti, J. P., "The long road to WiMAX," *IEE Review*, pp. 39 – 42, October 2005.

[13] Elson, J. and Estrin, D., "Time Synchronization for Wireless Sensor Networks," *Proceedings of IEEE IPDPS Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing 2001*, pp. 1965 – 1970. April 2001.

[14] Eng, C. S., *Digital Antenna Architectures using Commercial Off the Shelf Hardware*, Master's Thesis, Naval Postgraduate School, Monterey, CA., December 2003.

[15] Esswein, L. C., "Genetic algorithm design and testing of a random element 3-D 2.4 GHz phased array transmit antenna constructed of commercial RF microchips," Master's Thesis, Naval Postgraduate School, Monterey, CA., June 2003.

[16] Godara, L. C., "Application of antenna arrays to mobile communications, Part II: Beam-forming and direction-of-arrival considerations," *Proceedings of the IEEE*, Vol. 85, No. 8, pp. 1195 – 1245, 1997.

[17] Goodman, D. J., Valenzula, R. A., Gayliard, K. T., Ramamurthi, B., "Packet Reservation Multiple Access for Local Wireless Communications," *IEEE Transactions on Communications,* Vol. 37, No. 8, August 1989, pp. 885 – 890.

[18] Ino, Y., *A Study of High-Precision Array Antenna System for Adaptive Antenna in Mobile Communication*, Doctoral Dissertation, Graduate School of Engineering Yokohama National University, Japan.

[19] Janaswamy, R., *Radiowave Propagation and Smart Antennas for Wireless Communications,* Luwer Academic Publishers, Boston, 2001.

[20] Jenn, D. C., Program review, "Wirelessly networked opportunistic digital array radar," Naval Postgraduate School, January 2006 (unpublished).

[21] Jenn, D.C., Program review, *Aperstructures, Opportunistic Arrays and BMD*, Naval Postgraduate School, October 2005 (unpublished).

[22] Jiang, S., Rao, J., He, D., Ling, X., Ko, C. C., "A Simple Distributed PRMA for MANETs," *IEEE Transactions on Vehicular Technology,* Vol. 51, No. 2, March 2002, pp. 293 – 305.

[23] "Joint Vision 2020," US Government Printing Office, June 2000, http://www.dtic.mil/jointvision/jvpub2.htm, last accessed March 2005.

[24] Johnson, J. J., *Implementing the Cross Ambiguity Function and Generating Geometry-Specific Signals*, Master's Thesis, Naval Postgraduate School, Monterey, CA., September 2001.

[25] Jurdak, R., Lopes, C. V., and Baldi, P., *A Survey, Classification and Comparative Analysis of Medium Access Control Protocols For Ad Hoc Networks*, University of California, Irvine.

[26] Koushanfar, F., Slijepcevic, S., Potkonjak, S. M. and Sangiovanni-Vincentelli, A., "Location Discovery in Ad-hoc Wireless Sensor Networks," in *Ad Hoc Wireless Network,* Xiuzhen Cheng, Xiao Huang and Ding-Zhu Du, pp. 137 – 173, Kluwer Academic Pub., N. Y. 2004.

[27] Litva, J. and Lo, T. K. Y., *Digital Beamforming in Wireless Communications*, Artech House, Norwood, MA., 1996.

[28] Loke, Y., *Sensor Synchronization, Geolocation and Wireless Communication in a Shipboard Opportunistic Array*, Master's Thesis, Naval Postgraduate School, Monterey, CA., March 2006.

[29] Loomis, Jr. H. H., "Geolocation of Electromagnetic Emitters," NPS-EC-00-003, Naval Postgraduate School, Monterey, CA., Revised October 2003.

[30] Madisetti, V. K. and Williams, D. B., The Digital Signal Processing Handbook, CRC Press in cooperation with IEEE Press, 1998.

[31] Mailloux, R. J., *Phased Array Antenna Handbook*, pp. 29 – 37, Artech House, Boston, 1994.

[32] Miller, C. M., *Mitigating Systematic Error in Geolocation*, Doctoral Dissertation, Naval Postgraduate School, Monterey, CA., June 2005.

[33] Miller, M. I. and Fuhrmann, D. R., "Maximum likelihood narrow-band direction finding and the EM algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1560–1577, 1990.

[34] Miroslav, D. L., Milan, J. L., and Borislav, O. L., "Analysis of SDMA & smart antenna techniques for existing and new mobile communication systems," http://www.telfor.org.yu/telfor2001/radovi/4-1.pdf, November 2003.

[35] Murthy, C. and Manoj, B. S., *Ad Hoc Wireless Networks Architectures and Protocols*, pp. 647 – 693, Prentice Hall, Upper Saddle River N. J., 2004.

[36] Nasipuri, A. and Li, K., "A Directionality-Based Localization Scheme for Wireless Sensor Networks," *Proceedings of Association for Computing Machinery Workshop on Wireless Sensor Networks and Applications 2002*, pp. 105 – 111, September 2002.

[37] "Navy Concept Development and Experimentation Expeditionary Power Projection," June 2001, http://www.dtic.mil/ndia/2001ewc/ncde.pdf, last accessed March 2005.

[38] Niculescu, D., "Positioning in ad hoc sensor networks," IEEE Network, vol. 18, no. 4, pp. 24–29, July – August 2004.

[39] Ofek, Y., "Generating a Fault-Tolerant Global Clock Using High-Speed Control Signals for the MetaNet Architecture," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 1650 – 1654, 2002.

[40] Ong, C. S., *Digital Phased Array Architectures for Radar and Communications Based on Off-the-shelf Wireless Technologies*, Master's Thesis, Naval Postgraduate School, Monterey, CA., December 2004.

[41] Patwari, N., *Location Estimation in Sensor Networks*, Doctoral Dissertation, University of Michigan, 2005.

[42] Price, M. G., "Mathematics of Geolocation," Technical Memorandum, 1994.

[43] Rabideau, D. J., "Improved wideband time delay beam-steering," *Thirty-fifth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 1385 – 1390, 2001.

[44] Razavilar, J., Rashid-Farrokhi, F., and Liu, K. J. R., "Software radio architecture with smart antennas: a tutorial on algorithms and complexity," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 4, pp. 662 – 676, 1999.

[45] Roy, S., Foerster, J. R., Somayazulu, V. S. and Leeper, D. G., "Ultrawideband radio design: The promise of high-speed, short-range wireless connectivity," *Proceedings of the IEEE*, Vol. 92, No. 2, pp. 297, February 2004.

[46] Savvides, A., Girod, L, Srivastava, M. B., and Estrin, D., "Localization in sensor networks," in Wireless Sensor Networks, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Kluwer Academic Publishers, 2004.

[47] Schmidt, R. O., "Multiple emitter location and signal parameter estimation," *Proceedings, RADC Spectral Estimation Workshop*, pp. 243 – 258, October 1979.

[48] Schmidt, R. O. and Franks, R. E., "Multiple source df signal processing: An experimental system", in IEEE Transactions on Antennas and Propagation, March 1986, vol. AP-34, pp. 281 – 290.

[49] Scolnik, D. P. and Coleman, J. O., "Optimal design of wideband array patterns," *IEEE International Radar Conference*, pp. 172 – 177, 2000.

[50] Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., and Chandrakasan, A., "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," *Proceedings of Association for Computer Machinery Mobile Computing and Networking 2001*, pp. 272 – 286, July 2001.

[51] Shukla, D., Chandran-Wadia, L. and Sridhar Iyer, K. R., *Mitigating the Exposed Node Problem in IEEE 802.11 Ad Hoc Networks*, School of Information Technology Indian Institute of Technology Bombay.

[52] Sivalingam, K. M., "Tutorial on Wireless Sensor Network Protocols," *International Conference on High-Performance Computing 2002*, Bangalore, India, December 2002.

[53] Skolnik, M. I., *Introduction to Radar Systems*, 3rd edition, McGraw-Hill, N. Y., pp. 636, 2001.

[54] Sousa, E. S., Silvester J. A., "Spreading Code Protocols for Distributed Spread-Spectrum Packet Radio Networks," *IEEE Transactions on Communications,* Vol. 36, No. 3, 1973, pp. 272 – 281.

[55] Stallings, W., *Data & Computer Communications*, Fifth Edition, Prentice Hall, Upper Saddle River, N. J., 1996.

[56] Stein, S., Algorithms for ambiguity function processing, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 3, pp. 588 – 599, 1981.

[57] Steinberg, B. D., *Principles of Aperture and Array System Design, including Random and Adaptive Arrays*, John Wiley & Sons, N. Y., 1976.

[58] Stoica, P. and Moses, R., *Introduction to Spectral Analysis*. Prentice Hall, Upper Saddle River, N. J., 1997.

[59] Stoica, P. and Sharman, K. C., "Maximum likelihood methods for direction of arrival estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 1132–1143, 1990.

[60] Stutzman, W. L. and Thiele, G. A., Antenna Theory and Design, John Wiley and Sons, N. Y., 1981.

[61] Sun, G., Chen, J., Guo, W., and Liu, K. J. R., "Signal processing techniques in network-aided positioning," IEEE Signal Processing Mag., vol. 22, no. 4, pp. 12–23, July 2005.

[62] Svantesson, T., *Antennas and Propagation from a Signal Processing Perspective*, Doctoral Dissertation, Chalmers University of Technology, Sweden 2001.

[63] Taheri, S. H. and Steinberg, B. D., "Tolerance in self-cohering antenna arrays of arbitrary geometry," *IEEE Transactions on Antennas and Propagation*, Vol. AP-24, No. 5, pp. 733 – 739, September 1976.

[64] Tarran, C., Mitchell, M., and Howard, R., "Wideband phased array radar with digital adaptive beamforming," *High Resolution Radar and Sonar*, IEE Colloquium, pp. 1/1 – 1/7, 1999.

[65] Tingle, M. E., *Performance Evaluation of a Prototyped Wireless Ground Sensor*, Master's Thesis, Naval Postgraduate School, Monterey, CA., March 2005

[66] Therrien, C. W., *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall, Englewood Cliffs, N. J., 1992.

[67] Torrieri, D. J., "Statistical theory of passive location systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-20, No. 2, pp. 183 – 198, 1984.

[68] Tummala, M., Wai, C. C., Vincent, P., *Distributed Beamforming in Wireless Sensor Networks*, Conference Record of the Thirty-ninth Asilomar Conference on Signals, Systems & Computers, October 28 – November 01, 2005.

[69] Van Trees, H. L., *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, N. Y., 2002.

[70] Van Veen, B. D. And Buckley, K. M.. "Beamforming: A Versatile Approach to Spatial Filtering". *Signal Proc. Magazine*, pages 4–24, April 1988.

[71] Varma, K., *Time-Delay-Estimate Based Direction-of-Arrival Estimation for Speech in Reverberant Environments*, Master's Thesis, Virginia Polytechnic Institute, Blacksburg, VA.

[72] Whalen, A. D., McDonough, R.N., *Detection of Signals in Noise*, 2nd Edition, Academic Press, 1995.

[73] Wai, C. C., *Distributed Beamforming in Wireless Sensor Networks*, Master's Thesis, Naval Postgraduate School, Monterey, CA., December 2004.

[74] Wireless LAN MAC and Physical Layer Specifications, IEEE 802.11, 1999 Available: http://standards.ieee.org/getieee802/802.11.html last accessed April 2006.

[75] Woo, A. and Culler, D., "A Transmission Control Scheme for Media Access in Sensor Networks," *Proceedings of Association for Computer Machinery Mobile Communications and Networking 2001*, pp. 221 – 235, July 2001.

[76] Zeadally, S. and Zhang, L., "Enabling gigabit network access to end users," *Proceedings of the IEEE*, Vol. 92, No. 2, pp. 340 – 353, February 2004.

APPENDIX A. BRUTE_FORCE

```
% This is the brute_force program for beam force technique.
%
% Can use either stored Array or create an array on the fly, position
of 400 elements and M*N1=400;

clear
%load('Array');
M=20;                 %size(x_err);
N1 = M;
L = 500;
error2= 0.0;
[x,x_err,y,y_err,distance,midpointX,midpointY]=Random_LOC_DOA(M,N1,L,er
ror2);
    %x=x_err;
    %y=y_err;
    figure(1), plot(x,y,'*'), grid on, hold on
N=M*N1;
x_err1=reshape(x_err,N,1)';
y_err1=reshape(y_err,N,1)';

fc=2400e6;k=2*pi/(3e8/fc);rad=pi/180;                          % element
frequency and constants
r=sqrt(sum([x_err1.*x_err1;y_err1.*y_err1]));[min_r    ref]=min(r);    %
select reference element closest to origin
phase_ref=mod(k*2*(r-min_r)/rad,360);                         % generate
reference phase
phase=zeros(1,N);                                            % initialize
phase-shifter
%amp=1./(r/min_r);
amp=ones(1,N);
time=1;                                                       %initialize
time tag
for i=1:N
    element_start(i)=time;
    error(time)=amp(i)*exp(j*(phase(i)*rad))*exp(j*k*2*r(i))-
exp(j*k*2*r(ref));
    errorp(time)=mod((phase(i)*rad+k*2*r(i)-k*2*r(ref))/rad,360);
    if errorp(time)>180,
        errorp(time)=errorp(time)-360;
    end
     while ~((abs(error(time))<=0.2)||(time-element_start(i))>14)
        phase(i)=phase(i)+22.5;
        time=time+1;
        error(time)=amp(i)*exp(j*(phase(i)*rad))*exp(j*k*2*r(i))-
exp(j*k*2*r(ref));
        errorp(time)=mod((phase(i)*rad+k*2*r(i)-k*2*r(ref))/rad,360);
        if errorp(time)>180,
            errorp(time)=errorp(time)-360;
        end
    end
    time=time+1;
end
figure(2)                                  %generate plot
```

```
hold on, grid on
for i=1:N
    a=element_start(i);
    if i==N,
        b=length(error);
    else
        b=element_start(i+1)-1;
end
    if rem(i,7)==1,
        plot((a:b),errorp(a:b),'b')
    elseif rem(i,7)==2,
        plot((a:b),errorp(a:b),'g')
    elseif rem(i,7)==3,
        plot((a:b),errorp(a:b),'r')
    elseif rem(i,7)==4,
        plot((a:b),errorp(a:b),'c')
    elseif rem(i,7)==5,
        plot((a:b),errorp(a:b),'m')
    elseif rem(i,7)==6,
        plot((a:b),errorp(a:b),'y')
    else
        plot((a:b),errorp(a:b),'k')
    end
end
xlabel('Number of iterations')
ylabel('Phase error (deg)')
figure(3)
final_errorp=mod(phase_ref+phase,360);
for i=1:length(final_errorp)
    if final_errorp(i)>180,
        final_errorp(i)=final_errorp(i)-360;
    end
end
plot(final_errorp,'xr')
xlabel('Element number')
ylabel('Final Phase error (deg)')
grid on
axis([0 500 -40 40])
```

APPENDIX B. BEAM_TAGGING

```
% This is the beam_tag1 program for beam tagging technique.
%
% Used stored Array created for comparison with brute_force, position
of 400 elements and M*N1=100;

clear
load('Array');
M=size(x_err);
N=prod(M);
x_err1=reshape(x_err,N,1)';
y_err1=reshape(y_err,N,1)';

fc=3e8;k=2*pi/(3e8/fc);rad=pi/180;                              % element
frequency and constants
r=sqrt(sum([x_err1.*x_err1;y_err1.*y_err1]));[min_r    ref]=min(r);    %
select reference element closest to origin
phase_ref=mod(k*2*(r-min_r)/rad,360);                         % generate
reference phase
phase=zeros(1,N);                                            % initialize
phase-shifter

% Effects of amplitude with distance
%amp=1./(r/min_r);
amp=ones(1,N);

time=1;                                                       %initialize
time tag
command=[2];
for i=1:N
    element_start(i)=time;
    command=[command 0];
    while command(length(command))+command(length(command)-1)~=0,

F_pos(time)=amp(i)*exp(j*((phase(i)+90)*rad+k*2*r(i)))+exp(j*k*2*r(ref)
);
        F_neg(time)=amp(i)*exp(j*((phase(i)-
90)*rad+k*2*r(i)))+exp(j*k*2*r(ref));
        if abs(F_pos(time))<abs(F_neg(time)),
            phase(i)=phase(i)-22.5;command=[command -1];
        else
            phase(i)=phase(i)+22.5;command=[command 1];
        end
        errorp(time)=mod((phase(i)*rad+k*2*r(i)-k*2*r(ref))/rad,360);
        if errorp(time)>180,
            errorp(time)=errorp(time)-360;
        end
        time=time+1;
    end
end
figure(1)                                  %generate plot
hold
for i=1:N
    a=element_start(i);
```

```
    if i==N,
        b=length(errorp);
    else
    b=element_start(i+1)-1;
end
    if rem(i,7)==1,
        plot((a:b),errorp(a:b),'b')
    elseif rem(i,7)==2,
        plot((a:b),errorp(a:b),'g')
    elseif rem(i,7)==3,
        plot((a:b),errorp(a:b),'r')
    elseif rem(i,7)==4,
        plot((a:b),errorp(a:b),'c')
    elseif rem(i,7)==5,
        plot((a:b),errorp(a:b),'m')
    elseif rem(i,7)==6,
        plot((a:b),errorp(a:b),'y')
    else
        plot((a:b),errorp(a:b),'k')
    end
end
xlabel('Number of iterations')
ylabel('Phase error (deg)')
figure(2)
final_errorp=mod(phase_ref+phase,360);
for i=1:length(final_errorp)
    if final_errorp(i)>180,
        final_errorp(i)=final_errorp(i)-360;
    end
end
plot(final_errorp,'xr')
xlabel('Element number')
ylabel('Steady Phase error (deg)')
grid
axis([0 500 -90 90])
%figure(7)
%for i=1:N
%    if i==N,
%        count_tag(i)=time-element_start(i);
%    else
%    count_tag(i)=element_start(i+1)-element_start(i);
%end
%end
%plot(count_tag-1,'xr')
%xlabel('Element Number')
%ylabel('Number of iterations required')
%figure(8)
%hist(count_tag-1)
%xlabel('Number of iterations required')
%ylabel('Number of elements')
```

APPENDIX C. CAF I

```
function [TDOA, FDOA] = CAF(S1, S2, Max_f, fs, Max_t);
% ********************************************************************
% CAF takes as inputs two sampled signal vectors (S1 & S2) in analytic
% signal format, the maximum expected FDOA in Hertz (Max_f), the
% sampling frequency used to generate S1 & S2 (fs), and the maximum
% expected TDOA in seconds (Max_t). The function then utilizes
% Stein's method in [1] to compute coarse estimations of TDOA and
% FDOA between S1 & S2. Finally, "fine mode" calcualtions are made
% to compute the final TDOA and FDOA, which are returned to the
% user via the output arguments.
% Written by: LCDR Joe J. Johnson, USN
% Last modified: 17 September 2001
% Modified by CDR Mickey Batson, USN
% Last modified: 01 October 2006
% ********************************************************************
clc;
N = length(S1);
S1 = reshape(S1,N,1); % Ensures signals are column vectors due to
S2 = reshape(S2,N,1); % Matlab's better efficiency on columns
S1_orig = S1; % Want to preserve original input signals
S2_orig = S2; % for later use; S1 & S2 will be
% manipulated in the fine mode below.
% The following while loop ensures that the sub-block size, N1, is
% large enough to ensure proper resolution. If Max_f/fs*N1 were
% less than 1, then the Freq calculated at the end would always be
% + or – 1/N1! 2^19 = 524288 is about the limit for efficient
% processing speed.
N1=1024;
while (Max_f/fs*N1 < 2) & (N1 < 2^19)
N1 = 2*N1;
end
N2=N1/2;

if N1 > N % For cases where resolution calls for
S1 = [S1;zeros(N1-N,1)]; % a sub-block size larger than the
S2 = [S2;zeros(N1-N,1)]; % signal vectors, pad the vectors with
N = N1; % zeros so that they have a total of
end % N1 elements.
% Want magnitude of Max_f, since +&- will be used below
Max_f = abs(Max_f) ;
Number_of_Blocks = length(S1)/N1; % Number of sub-blocks to break
% the signal into
Min_v = floor(-Max_f/fs*N1); % Smallest freq bin to search
Max_v = -Min_v; % Largest freq bin to search
v_values = Min_v : Max_v; % Vector of all bins to search
Max_samples = Max_t * fs; % Maximum number of samples to search
% Finds max number of block shifts (q) that must occur for each
% R and v below.
If Max_samples > N2
q_max = min(ceil((Max_samples – N2)/N1),Number_of_Blocks-1);
else q_max = 0;
end
x=0;
```

```
divisors = Number_of_Blocks:-1:1; % Used to scale “temp” below…
% ********************************************************************
% COARSE MODE computations.
% ********************************************************************
for v = 1:length(v_values)
temp(1:N1,1:q_max+1)=0; % Initializing – saves time….
for R = 0:Number_of_Blocks-1
% temp1 is the FFT of the R’th block of S1, shifted by “v” bins.
Temp1 = fftshift(fft(S1(1+R*N1 : N1*(R+1))));
temp1 = shiftud(temp1,v_values(v),0);
for q = 0:q_max
% R+q cannot exceed the number of sub-blocks
if R + q > Number_of_Blocks-1 break
end

% FFT of the (R+q)’th block of S2
temp2 = fftshift(fft([S2(1+(R+q)*N1 : N2 + N1*(R+q));…
zeros(N2,1)]));
% Multiplies temp1 & temp2, FFTs the product, then adds to
% previous values for the same value of q (but different R)
temp(:,q+1) = temp(:,q+1) + …
abs(fftshift(fft(temp1.*conj(temp2))));
end
end
% Each value of q was used a different # of times, so they must be
% scaled properly.
For q_index = 1:q_max+1
temp(:,q_index) = temp(:,q_index) / divisors(q_index);
end
% If combination of current v and any q provides a greater value
% than the previous max, then remember m, Q, & V.
if max(max(temp))>x
x = max(max(temp));
[m Q] = find(temp == max(max(temp)));
% Must do this since q starts at 0, but Matlab doesn’t allow for
% zero indexing.
Q = Q – 1;
V = v_values(v);
end
end
% Coarse estimate of TDOA (in # of samples)
TDOA_Coarse = Q * N1 + (-N2+1 + m) ;
% Coarse estimate of FDOA (in Freq Bin #)
FDOA_Coarse = V/N1*N;
% The following 3 lines can be used to display the coarse estimates,
% if desired.
%disp([‘The coarse TDOA estimate is: ‘, num2str(TDOA_Coarse),…
% ‘ samples.’]);
%disp([‘The coarse FDOA estimate is: ‘, num2str(FDOA_Coarse/N),…
% ‘ (digital frequency).’]);

% ********************************************************************
% FINE MODE computations.
% ********************************************************************
S2 = conj(S2); % S2 is conjugated in basic CAF definition
% Vector of freq “bins” to use (DON’T have to be integers!!)
k_val = FDOA_Coarse-10 : FDOA_Coarse+10;
```

```
% Vectors of TDOAs to use (must be integers)
tau_val = TDOA_Coarse-10 : TDOA_Coarse+10 ;
done = 0;
multiple = 1;
decimal = 0;
while ~done % Fine mode iterations continue until user is done.
% Initialize to make later computations faster
amb(length(k_val),length(tau_val))=0;
Ntemp = N * multiple;
for k = 1:length(k_val) % Must loop through all values of k
% Vector of complex exponentials that will be used
exponents = exp(-j*2*pi*k_val(k)/Ntemp*(0 :Ntemp-1)') ;
% Must loop through all potential TDOAs
for t = 1:length(tau_val)
% S2 is shifted "tau" samples
S2temp = shiftud(S2,tau_val(t),0);
% Definition of CAF summation
temp = abs(sum(S1.*S2temp.*exponents));
% Save CAF magnitude for the values of k & t
amb(k,t)=temp;
end
end
[k, t]=find(amb==max(max(amb))); % Find the peak of the CAF matrix
TDOA = tau_val(t); % TDOA and FDOA associated with the peak of the
FDOA = k_val(k); % CAF plane. These represent the final TDOA
% & FDOA estimates.

% The results are displayed.
Disp(' ');disp(' ');disp(' ');
disp(['The TDOA is ', num2str(TDOA/multiple), ' samples']);
disp([' or ', num2str(TDOA/(multiple*fs)), ' seconds.']);
disp(' ');
disp(['The resolution is ', num2str(0.5/…
(multiple*fs)),' seconds.']) ;
disp(' ');disp(' ');
disp(['The FDOA is ', num2str(FDOA/N),…
' in digital frequency (k/N)']);
disp([' or ', num2str(FDOA/N*fs), ' Hz.']); disp(' ');
disp(['The resolution is ', num2str(0.5*…
(10^decimal)/N*fs), ' Hz.']);
disp(' ');disp(' ');disp(' ');
% If the signal length exceeds 524288 elements, max processing
% capability has been achieved, and the user will not be given
% the option of refining TDOA any further.
If Ntemp >= 2^19
disp('Maximum TDOA processing capability has been achieved.')
doneT = 1;
else doneT = 0;
end
% User chooses whether to compute more accurate TDOA &/or
% FDOA, or to stop fine mode computations.
Disp('Do you desire a solution with finer resolution?');
disp('Select one of the following:'); disp(' ');
if ~doneT
disp('1. Finer resolution for TDOA.');
else disp(' ');
end
```

```
disp('2. Finer resolution for FDOA.');
if ~doneT
disp('3. Finer resolution for both TDOA and FDOA.');
else disp(' ');
end
disp('4. The TDOA and FDOA resolutions are fine enough.');
disp(' ');

choice = input('What is your selection? ');
switch choice
% TDOA is refined by resampling the signals at twice the
% previous sampling rate. Increases resolution two-fold.
Case 1
if ~doneT
multiple = multiple*2;
S1 = interp(S1, 2);
S2 = interp(S2, 2);
tau_val = TDOA*2 – 1 : TDOA*2 + 1;
else done = 1;
end
clc;
% FDOA resolution is improved by a factor of 10.
Case 2
decimal = decimal – 1;
k_val = FDOA – 5*10^decimal : 10^decimal : FDOA + 5*10^decimal;
clc;
% Both TDOA and FDOA resolutions are improved.
Case 3
if ~doneT
multiple = multiple*2;
S1 = interp(S1, 2);
S2 = interp(S2, 2);
tau_val = TDOA*2 – 1 : TDOA*2 + 1;
decimal = decimal – 1;
k_val = FDOA – 5*10^decimal : 10^decimal : FDOA + …
5*10^decimal;
else done = 1;
end
clc;
otherwise
done = 1;
end
if done
disp(' ');disp(' '); disp('TDOA & FDOA estimation complete.');
end
end

% If user wants to see the CAF surface graphically, a call to
% CAF_peak is made.
Disp(' ');%disp(' ');disp(' ');
choice = input…
('Would you like to see the CAF peak graphically (Y or N)? ','s');
choice = upper(choice);
switch choice
case 'Y'
caf_peak(S1_orig, S2_orig, floor(TDOA/multiple) – 50, …
floor(TDOA/multiple) + 50, (FDOA-20)/N, (FDOA+20)/N, fs);
```

```
end
TDOA = TDOA/(multiple*fs); % Returns TDOA in seconds.
FDOA = FDOA/N*fs; % Returns FDOA in Hertz.
Disp('Program Complete.');
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. CAF II

```
function [TDOA, FDOA, MaxAmb, Amb] = ...
CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, fs);
% ********************************************************************
% CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi) takes as input:
% two signals (S1, S2) that are row or column vectors; a range of
% time delays (in samples) to search (Tau_Lo, Tau_Hi must be
% integers between -N & +N); a range of digital frequencies (in
% fractions of sampling frequency) to search (Freq_Lo, Freq_Hi must
% be between -1/2 and 1/2, or -(N/2)/N and (N/2)/N, where N is the
% length of the longer of the two signal vectors); and the sampling
% frequency, fs.
%
% The function computes the Cross Ambiguity Function of the two
% signals. Four plots are produced which represent four different
% views of the Cross Ambiguity Function magnitude versus the input
% Tau and Frequency Offset ranges.
%
% The function returns the scalars TDOA, FDOA, and MaxAmb, where
% TDOA & FDOA are the values of Time Delay and Frequency Offset
% that cause the Cross Ambiguity Function to peak at a magnitude
% of MaxAmb. Amb is the matrix of values representing the CAF
% surface.

% Written by: LCDR Joe J. Johnson, USN
% Last modified: 26 August 2001
% Modified by CDR Mickey Batson, USN
% Last modified: 10 October 2006
% ********************************************************************
% Ensures that the user enters all SIX required arguments.
if (nargin < 6)
error...
('6 arguments required: S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi');
end
% Ensures that both S1 & S2 are row- or column-wise vectors.
if ((size(S1,1)~=1)&(size(S1,2)~=1)) | ((size(S2,1)~=1)&...
(size(S2,2)~=1))
error('S1 and S2 must be row or column vectors.');
end
N1 = length(S1);
N2 = length(S2);
S1 = reshape(S1,N1,1); % S1 & S2 are reshaped into column-wise
S2 = reshape(S2,N2,1); % vectors since MATLAB is more efficient
% when manipulating columns.
S1 = [S1;zeros(N2-N1,1)]; % Ensure that S1 & S2 are the same size,
S2 = [S2;zeros(N1-N2,1)]; % padding the smaller one w/ 0s as neeeded.
% This WHILE loop simply ensures that the length of S1 & S2 is a power
% of two. If not, the vectors are padded with 0s until their length
% is a power of two. This is not required, but it takes advantage of
% the fact that MATLAB's FFT computation is significantly faster for
% lengths which are powers of two!
while log(length(S1))/log(2) ~= round(log(length(S1))/log(2))
S1(length(S1)+1) = 0;
```

```
S2(length(S2)+1) = 0;
end
N = length(S1);
% Ensures that the Tau values entered are in the valid range.
if abs(Tau_Lo)>N | abs(Tau_Hi)>N
error('Tau_Lo and Tau_Hi must be in the range -N to +N.');
end

% Ensures that Tau values entered by the user are integers.
if (Tau_Lo ~= round(Tau_Lo)) | (Tau_Hi ~= round(Tau_Hi))
error('Tau_Lo and Tau_Hi must be integers.')
end
% Ensures that the Frequency values entered are in the valid range.
if abs(Freq_Lo)>1/2 | abs(Freq_Hi)>1/2
error('Freq_Lo and Freq_Hi must be in the range -.5 to +.5');
end
% Ensures that the lower bounds are less than the upper bounds.
if (Tau_Lo > Tau_Hi) | (Freq_Lo > Freq_Hi)
error('Lower bounds must be less than upper bounds.')
end
% Freq values converted into integers for processing.
Freq_Lo = round(Freq_Lo*N);
Freq_Hi = round(Freq_Hi*N);
% Creates vectors for the Tau & Freq values entered by the user. Used
% for plotting...
TauValues = [Tau_Lo:Tau_Hi];
FreqValues = [Freq_Lo:Freq_Hi]/N;
% The IF statement calculates the indices required to isolate the
% user-defined frequencies from the FFT calculations below.
if Freq_Lo < 0 & Freq_Hi < 0
Neg_Freq = (N+Freq_Lo+1:N+Freq_Hi+1);
Pos_Freq = [];
elseif Freq_Lo < 0 & Freq_Hi >= 0
Neg_Freq = (N+Freq_Lo+1:N);
Pos_Freq = (1:Freq_Hi+1);
else
Neg_Freq = [];
Pos_Freq = (Freq_Lo+1:Freq_Hi+1);
end
% This FOR loop actually calculates the Cross Ambiguity Function for
% the given range of Taus and Frequencies. Note that an FFT is
% performed for each Tau value and then the frequencies of interest
% are isolated using the Neg_Freq and Pos_Freq vectors obtained above.
% For each value of Tau, the vector S2 is shifted Tau samples using a
% call to the separate function "SHIFTUD". Samples shifted out are
% deleted and zeros fill in on the opposite end.

% Initializing Amb with 0s makes computations much faster.
Amb=zeros(length(Neg_Freq)+length(Pos_Freq),length(TauValues));
for t = 1:length(TauValues)
temp = fft((S1).*conj(shiftud(S2,TauValues(t),0)));
Amb(:,t) = [temp(Neg_Freq);temp(Pos_Freq)];
end
% Only interested in the Magnitude of the Cross Ambiguity Function.
Amb = abs(Amb);
% The following will remove any spike that occurs at Tau = FreqOff = 0.
% This may be desired in some cases, especially when the spike at (0,0)
```

```
% is due to correlation of the two signals' noise components. The
% spike, of course, could also indicate that the two signals have no
% TDOA or FDOA between them.
% if find(TauValues == 0) & find(FreqValues == 0)
% Amb(find(FreqValues==0),find(TauValues==0)) = 0;
% end
%clc; %Clears the MATLAB command window.
% The four different views of the Cross Ambiguity Function plots are
% created here.
figure % This one is the 3-D view
mesh(TauValues/fs,FreqValues*fs,Amb);
xlabel('TDOA (Seconds)');ylabel('FDOA (Hertz)');
zlabel('Magnitude');
title('Cross Ambiguity Function');
figure
subplot(2,1,1) % This one is the 2-D view along the TDOA axis
mesh(TauValues/fs,FreqValues*fs,Amb);
xlabel('TDOA (Seconds)');
zlabel('Magnitude');
view(0,0);
subplot(2,1,2) % This one is the 2-D view along the FDOA axis
mesh(TauValues/fs,FreqValues*fs,Amb);
ylabel('FDOA (Hertz)');
zlabel('Magnitude');
view(90,0);

%This one is a 2-D view looking down on the plane
figure
mesh(TauValues/fs,FreqValues*fs,Amb);
xlabel('TDOA (Seconds)');ylabel('FDOA (Hertz)');
zlabel('Magnitude');
title('Cross Ambiguity Function');
view(0,90);
% Finds the indices of the peak value.
[DFO, DTO] = find(Amb==max(max(Amb)));
TDOA = TauValues(DTO); % Finds the actual value of the TDOA.
FDOA = FreqValues(DFO); % Finds the actual value of the FDOA.
MaxAmb = max(max(Amb)); % Finds the actual Magnitude of the peak.
% The remaining lines will display the numerical results of the
% TDOA & FDOA, if desired. Since the FFT method was used for the
% calculations, the TDOA is accurate only to within +/- 0.5 samples,
% and the FDOA is accurate to within +/- 0.5/N in digital frequency.
% disp(' '); disp(' ');
% disp(['The TIME LAG (TDOA) is: ',num2str(TDOA),' Samples.']);
% disp(' ');
% disp(['The FREQ OFFSET (FDOA) is: ',num2str(FDOA),...
% ' (Fraction of Fs).']);
% disp(' '); disp(['Maximum Magnitude = ',num2str(MaxAmb)]);
% disp(' ');disp('-----------------------------');
% disp('NOTE: If the CAF plot has secondary peaks whose magnitudes');
% disp(' are within about 80% of the Main Peak''s magnitude,');
% disp(' then the above results may be unreliable. Likely');
% disp(' reasons: The true peak is not within the range of,');
% disp(' Taus & Freq Offsets that you entered or the signals');
% disp(' may be too noisy to detect the peak.');
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. TDOA MAIN

```
%  ---------------------------------------------------------------------
--------------------
% Filename: Random_MAIN.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
% error = position errors in meters (assumed uniformly distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% AF_mag = array factor
%  ---------------------------------------------------------------------
--------------------

%function[]=Random_MAIN()

clear
global c
Na=0;      %input('Enter number of random array samples to be generated
:');
M=2;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
sigm = 1e-9;
k = 5;
kk = M*N - 1;
z = [5e3;5e3];


wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=50;           %input('Enter length of sensor cluster square area in
meters :');
error= 0.0;   %input('Enter position error in meters :');
fail=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');

p=1;
    %load('Array');
    [x_out,x_err,y_out,y_err,distance,midpointX,
midpointY]=Loomis_LOC_DOA(M,N,L,error);
    %distance_lamda = distance./wavelength;
```

```
figure (1), plot(x_err, y_err, '*r'), grid on, hold on

vz = [0.0;0.0];
dt = 1e-6;
zi = [2e3;1e3];
for i = 1:9
zco(:,i) = [midpointX(i); midpointY(i)];
end
%D = reshape(D,M*N,1);
%[peaks, locs] = pkpick(D,Th,(M*N-accuracy));
%ZZC1 = zeros(2,k);%(M*N-accuracy));
%ZZC2 = zeros(2,k);%(M*N-accuracy));
%MMM = zeros(1,(M*N-accuracy));
%for i = 1:length(peaks)
%peaks = sort(peaks);
[m, zzc1, zzc2] = tdoagen(sigm, k, z, zco(:,2), distance(:,2), vz, dt);
 %ZZC1(:,p) = zzc1;
 %ZZC2(:,p) = zzc2;
 %MMM(:,p) = m;
 %p=p+1;
%end
tic
[mi, zz, P] = nrtdoa(k,m',zzc1,zzc2,vz,sigm,zi);
time_elapsed = toc
%azimuth = atan2(zz(2),zz(1));
%Azimuth = azimuth*(180/pi)
[m2, zzc12, zzc22] = tdoagen(sigm, k, z, zco(:,3), distance(:,3), vz,
dt);
%compute and plot TDOA isochron contours
%lifted from geotdof2
%First set up extent of plot to include all features of interest
[mi2, zz2, P2] = nrtdoa(k,m2',zzc12,zzc22,vz,sigm,zi);

[m3, zzc13, zzc23] = tdoagen(sigm, k, z, zco(:,4), distance(:,4), vz,
dt);
[mi3, zz3, P3] = nrtdoa(k,m3',zzc13,zzc23,vz,sigm,zi);

[m4, zzc14, zzc24] = tdoagen(sigm, k, z, zco(:,5), distance(:,5), vz,
dt);
[mi4, zz4, P4] = nrtdoa(k,m4',zzc14,zzc24,vz,sigm,zi);

[m5, zzc15, zzc25] = tdoagen(sigm, k, z, zco(:,6), distance(:,6), vz,
dt);
[mi5, zz5, P5] = nrtdoa(k,m5',zzc15,zzc25,vz,sigm,zi);

[m6, zzc16, zzc26] = tdoagen(sigm, k, z, zco(:,7), distance(:,7), vz,
dt);
[mi6, zz6, P6] = nrtdoa(k,m6',zzc16,zzc26,vz,sigm,zi);

[m7, zzc17, zzc27] = tdoagen(sigm, k, z, zco(:,8), distance(:,8), vz,
dt);
[mi7, zz7, P7] = nrtdoa(k,m7',zzc17,zzc27,vz,sigm,zi);

[x,y]=meshgrid(-8e3:20:8e3,-8e3:20:8e3);
%[x,y]=meshgrid(-20e3:200:20e3,-20e3:200:20e3);
```

```
%figure (2), hold on, grid on

xlabel('x position in meters'), ylabel('y position in meters')
title('Line of Bearing')
%gtext('Sensor  Grid  2500  square  meters:  error  =  0.0  meter  &  node
failure = 0 nodes')

 for j=1:k
   % jth contour
  tb=(1/c).*(((x-(zzc1(1,j))).^2 + (y-(zzc1(2,j))).^2).^(.5) - ...
             ((x-(zzc2(1,j))).^2 + (y-(zzc2(2,j))).^2).^(.5));
  MT=max(max(tb));
  mt=min(min(tb));
  ttb=mi(j,1);
  tlb=[1 ttb];
  Tb=contour(-8e3:20:8e3,-8e3:20:8e3,tb,tlb,'b-');
  %Tb=contour(-20e3:200:20e3,-20e3:200:20e3,tb,tlb,'b-');
  ttb=mi(j,1)+sigm;
     tlb=[1 ttb];
     %Tb=contour(-8e3:20:8e3,-8e3:20:8e3,tb,tlb,'c-');
     %Tb=contour(-20e3:200:20e3,-20e3:200:20e3,tb,tlb,'c-');
     ttb=mi(j,1)-sigm;
     tlb=[1 ttb];
     %Tb=contour(-8e3:20:8e3,-8e3:20:8e3,tb,tlb,'c-');
     %Tb=contour(-20e3:200:20e3,-20e3:200:20e3,tb,tlb,'c-');

  tb2=(1/c).*(((x-(zzc12(1,j))).^2 + (y-(zzc12(2,j))).^2).^(.5) - ...
             ((x-(zzc22(1,j))).^2 + (y-(zzc22(2,j))).^2).^(.5));
  MT2=max(max(tb2));
  mt2=min(min(tb2));
  ttb2=mi2(j,1);
  tlb2=[1 ttb2];

   Tb2=contour(-8e3:20:8e3,-8e3:20:8e3,tb2,tlb2,'b-');
   %Tb2=contour(-20e3:200:20e3,-20e3:200:20e3,tb2,tlb2,'b-');
    ttb2=mi2(j,1)+sigm;
     tlb2=[1 ttb2];
     %Tb2=contour(-8e3:20:8e3,-8e3:20:8e3,tb2,tlb2,'c-');
     %Tb2=contour(-20e3:200:20e3,-20e3:200:20e3,tb,tlb,'c-');
     tb2=mi2(j,1)-sigm;
     tlb2=[1 ttb2];
     %Tb2=contour(-8e3:20:8e3,-8e3:20:8e3,tb2,tlb2,'c-');
     %Tb2=contour(-20e3:200:20e3,-20e3:200:20e3,tb,tlb,'c-');
  tb3=(1/c).*(((x-(zzc13(1,j))).^2 + (y-(zzc13(2,j))).^2).^(.5) - ...
             ((x-(zzc23(1,j))).^2 + (y-(zzc23(2,j))).^2).^(.5));
  MT3=max(max(tb3));
  mt3=min(min(tb3));
  ttb3=mi3(j,1);
  tlb3=[1 ttb3];
  Tb3=contour(-8e3:20:8e3,-8e3:20:8e3,tb3,tlb3,'b-');
  %Tb3=contour(-20e3:200:20e3,-20e3:200:20e3,tb3,tlb3,'b-');
  ttb3=mi3(j,1)+sigm;
     tlb3=[1 ttb3];
     %Tb3=contour(-8e3:20:8e3,-8e3:20:8e3,tb3,tlb3,'c-');
     %Tb3=contour(-20e3:200:20e3,-20e3:200:20e3,tb3,tlb3,'c-');
     ttb3=mi3(j,1)-sigm;
     tlb3=[1 ttb3];
```

```
   %Tb3=contour(-8e3:20:8e3,-8e3:20:8e3,tb3,tlb3,'c-');
   %Tb3=contour(-20e3:200:20e3,-20e3:200:20e3,tb3,tlb3,'c-');

tb4=(1/c).*(((x-(zzc14(1,j))).^2 + (y-(zzc14(2,j))).^2).^(.5) - ...
          ((x-(zzc24(1,j))).^2 + (y-(zzc24(2,j))).^2).^(.5));
MT4=max(max(tb4));
mt4=min(min(tb4));
ttb4=mi4(j,1);
tlb4=[1 ttb4];
Tb4=contour(-8e3:20:8e3,-8e3:20:8e3,tb4,tlb4,'b-');
%Tb4=contour(-20e3:200:20e3,-20e3:200:20e3,tb4,tlb4,'b-');
ttb4=mi4(j,1)+sigm;
   tlb4=[1 ttb4];
   %Tb4=contour(-8e3:20:8e3,-8e3:20:8e3,tb4,tlb4,'c-');
   %Tb4=contour(-20e3:200:20e3,-20e3:200:20e3,tb4,tlb4,'c-');
   ttb4=mi4(j,1)-sigm;
   tlb4=[1 ttb4];
   %Tb4=contour(-8e3:20:8e3,-8e3:20:8e3,tb4,tlb4,'c-');
   %Tb4=contour(-20e3:200:20e3,-20e3:200:20e3,tb4,tlb4,'c-');

tb5=(1/c).*(((x-(zzc15(1,j))).^2 + (y-(zzc15(2,j))).^2).^(.5) - ...
          ((x-(zzc25(1,j))).^2 + (y-(zzc25(2,j))).^2).^(.5));
MT5=max(max(tb5));
mt5=min(min(tb5));
ttb5=mi5(j,1);
tlb5=[1 ttb5];
Tb5=contour(-8e3:20:8e3,-8e3:20:8e3,tb5,tlb5,'b-');
%Tb4=contour(-20e3:200:20e3,-20e3:200:20e3,tb4,tlb4,'b-');
ttb5=mi5(j,1)+sigm;
   tlb5=[1 ttb5];
   %Tb5=contour(-8e3:20:8e3,-8e3:20:8e3,tb5,tlb5,'c-');
   %Tb5=contour(-20e3:200:20e3,-20e3:200:20e3,tb5,tlb5,'c-');
   ttb5=mi5(j,1)-sigm;
   tlb5=[1 ttb5];
   %Tb5=contour(-8e3:20:8e3,-8e3:20:8e3,tb5,tlb5,'c-');
   %Tb5=contour(-20e3:200:20e3,-20e3:200:20e3,tb5,tlb5,'c-');

tb6=(1/c).*(((x-(zzc16(1,j))).^2 + (y-(zzc16(2,j))).^2).^(.5) - ...
          ((x-(zzc26(1,j))).^2 + (y-(zzc26(2,j))).^2).^(.5));
MT6=max(max(tb6));
mt6=min(min(tb6));
ttb6=mi6(j,1);
tlb6=[1 ttb6];
Tb6=contour(-8e3:20:8e3,-8e3:20:8e3,tb6,tlb6,'b-');
%Tb6=contour(-20e3:200:20e3,-20e3:200:20e3,tb6,tlb6,'b-');
ttb6=mi6(j,1)+sigm;
   tlb6=[1 ttb6];
   %Tb6=contour(-8e3:20:8e3,-8e3:20:8e3,tb6,tlb6,'c-');
   %Tb6=contour(-20e3:200:20e3,-20e3:200:20e3,tb6,tlb6,'c-');
   ttb6=mi6(j,1)-sigm;
   tlb6=[1 ttb6];
   %Tb6=contour(-8e3:20:8e3,-8e3:20:8e3,tb6,tlb6,'c-');
   %Tb6=contour(-20e3:200:20e3,-20e3:200:20e3,tb6,tlb6,'c-');

tb7=(1/c).*(((x-(zzc17(1,j))).^2 + (y-(zzc17(2,j))).^2).^(.5) - ...
          ((x-(zzc27(1,j))).^2 + (y-(zzc27(2,j))).^2).^(.5));
MT7=max(max(tb7));
```

```
 mt7=min(min(tb7));
 ttb7=mi7(j,1);
 tlb7=[1 ttb7];
 Tb7=contour(-8e3:20:8e3,-8e3:20:8e3,tb7,tlb7,'b-');
 %Tb7=contour(-20e3:200:20e3,-20e3:200:20e3,tb7,tlb7,'b-');
 ttb7=mi7(j,1)+sigm;
    tlb7=[1 ttb7];
    %Tb7=contour(-8e3:20:8e3,-8e3:20:8e3,tb7,tlb7,'c-');
    %Tb7=contour(-20e3:200:20e3,-20e3:200:20e3,tb7,tlb7,'c-');
    ttb7=mi7(j,1)-sigm;
    tlb7=[1 ttb7];
    %Tb7=contour(-8e3:20:8e3,-8e3:20:8e3,tb7,tlb7,'c-');
    %Tb7=contour(-20e3:200:20e3,-20e3:200:20e3,tb7,tlb7,'c-');
End
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. TDOAGEN

```
function [tdoa,zzc1,zzc2] = tdoagen(sigm,k,z,zc0,D,vz,dt)
%  Generates k component vector of tdoas
% sigm – Standard deviation of time measurement
% k – number of measurements
% z – True position of emitter
% D – Distance separating collectors
% vz – Vector collector velocity
% dt – Time between TDOA measurements
% tdoa – k-component vector of simulated TDOA measurements
% Written by HH Loomis on 7/7/99
% Based on nrgeogrf.m version 3.41

% s – half separation
% zz – Matrix of collector positions, 2xk

global c
disp(‘ Generating’)
disp(k)
disp(‘ TDOA Observations.’)
s=D/2;
zz=zeros(2,k);

for j=1:k

%r1(:,j)=z-(zz(:,j)-[s;0]);
%r2(:,j)=z-(zz(:,j)+[s;0]);
%TDOA(j,1)=(1./c)*(norm(r1(:,j))-norm(r2(:,j))) + sigm*randn;
                  zz(:,j)=zc0; %+vz*(j-1)*dt;
                  zzc1(:,j)=zz(:,j)-zz(:,j); %[s;0];
                  zzc2(:,j)=zz(:,j)+zz(:,j)  %[s;0];
            tdoa(j,1)=(1./c)*(norm(z-(zzc1(:,j)))       –       norm(z-
(zzc2(:,j)))) + sigm*randn;

end
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. NEWTON-RAPHSON

```
function [mi, zz, P] = nrtdoa(k,m,zzc1,zzc2,vz,sigm,zi)
%function computes position estimate, zi, and covariance matrix, P
% vz - Vector collector velocity
% zzc1 - Matrix of collector 1 positions, 2xk
% zzc2 - Matrix of collector 2 positions, 2xk
% sigm - Standard deviation of time measurement
% zi - iterated estimated position of emitter
% Written by HH Loomis on 7/7/99
% Based on nrgeogrf.m version 3.41

% W - Conditioning matrix
% mi - vector of TDOAs for current zi
% A - kx2 PARTIAL DERIV. MATRIX,

global c

%     calculate Weighting Matrix, W

 %    limit=1.4*max(z);
 %     [x,y]=meshgrid(-limit:.02*limit:limit,-limit:.02*limit:limit);

 W=zeros(k,k);

 for jj=1:k
            W(jj,jj)=1/(sigm^2);
 end
% disp('W=' )
% disp(W)

%     start estimation loop with ii = 0

ii=0;
while ii<10
%  disp('Begin Iteration #:')
%  disp(ii)


%calculate mi as function of zi, initial trial value of position

  for jj=1:k


    mi(jj,1)=(1./c)*(norm(zi-(zzc1(:,jj))) - norm(zi-(zzc2(:,jj))));

  end

%     Compute rms error in mi

  dm=m'-mi;
  rms_m = norm(dm);

%   disp('mi, dm, rms_m')
%   disp(mi)
```

```
%   disp(dm)
%   disp(rms_m)
%   disp('mmi')
%   disp(mmi)

  if rms_m <= .1*sigm break
  else

%     CALCULATE PARTIAL DERIV. MATRIX, A(K,2)
    for jj=1:k
      A(jj,:)=(1/c)*(((zi-(zzc1(:,jj)))'/norm(zi ...
         -(zzc1(:,jj)))) - ...
       ((zi-(zzc2(:,jj)))'/norm(zi-(zzc2(:,jj)))));
    end
%    disp('Matrix A')
%    disp(A)
%disp ('Condition of At*W*A')
%cond(A'*W*A)

%     COMPUTE increment in z, dzi

    P=inv(A'*W*A);
    dzi=(P)*(A'*W)*dm;

% disp('dzi')
% disp(dzi)

%     COMPUTE next estimate for zi

    zip1=zi+dzi;
% disp('zip1')
% disp(zip1)

%ii used as loop index since i is sqrt(-1) in matlab

%    disp('end of iteration loop #')
%    disp(ii)
    ii=ii+1;
    zi=zip1;

  end %for if -- else

end %while ii
zz=zi
%end iteration loop

disp('ii')
disp(ii)
disp('zi, dzi')
disp(zi)
disp(dzi)
disp('dm')
disp(dm)
disp('rms_m')
disp(rms_m)
disp ('P')
disp (P)
```

APPENDIX H. RANDOM_LOC_DOA

```
%  ------------------------------------------------------------------------
---------------------
% Filename: Random_LOC.m
%  ------------------------------------------------------------------------
---------------------
% Input parameters:
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% L = length of square planar area (in meters)
% error = position errors (assumed uniformly distributed between [-
posterror and +posterror])
%  ------------------------------------------------------------------------
---------------------
% Output:
% x = x-coordinates of sensor nodes (in meters)
% x_err = x-coordinates of sensor nodes with errors (in meters)
% y = y-coordinates of sensor nodes (in meters)
% y_err = y-coordinates of sensor nodes with errors (in meters)
% distance = distance between the sensor nodes and the reference node
in
% meters
%  ------------------------------------------------------------------------
---------------------

function[x_loc,x_err,y_loc,y_err,distance,midpointX,midpointY]        =
Random_LOC(M,N,L,error,lamda)
%M=3;
%N=M;
%L=10;
%error=0;
%X = lamda.*[0 1 2 3 4; 5 6 7 8 9; 10 11 12 13 14; 15 16 17 18 19; 20
21 22 23 24];
%Y = lamda.*[0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4];
%Y = zeros(M,N);
x_loc=[0; -50; 30; 0; -30; 25; -40; -100; 100];
y_loc=[0; 0; -50; 70; -50; -25; 70; 50; -80];

%x_loc=L*X;
%y_loc=L*Y;

%x_loc(1,1)=zeros(1,1);
%y_loc(1,1)=zeros(1,1);
%dx=-error+(2*error)*rand(M,N);
%dy=-error+(2*error)*rand(M,N);
x_err=x_loc; %+dx;
y_err=y_loc; %+dy;

for i = 1:9
distance(i) = sqrt((x_err(1)-x_err(i))^2 + (y_err(1)-y_err(i))^2);
midpointX(i) = (x_err(1)+x_err(i))/2;
midpointY(i) = (y_err(1)+y_err(i))/2;
end
x=x_loc;
```

```
y=y_loc;
save ARRAY x y x_err y_err distance midpointX midpointY



%  ------------------------------------------------------------------------
----------
% Filename: Random_LOC_DOA.m
%  ------------------------------------------------------------------------
----------
% Input parameters:
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% L = length of square planar area (in meters)
% error = position errors (assumed uniformly distributed between [-
posterror and +posterror])
%  ------------------------------------------------------------------------
----------
% Output:
% x = x-coordinates of sensor nodes (in meters)
% x_err = x-coordinates of sensor nodes with errors (in meters)
% y = y-coordinates of sensor nodes (in meters)
% y_err = y-coordinates of sensor nodes with errors (in meters)
% distance = distance between the sensor nodes and the reference node
in
% meters
%  ------------------------------------------------------------------------
----------

function[x_loc,x_err,y_loc,y_err,distance,midpointX,midpointY]        =
Random_LOC_DOA(M,N,L,error,lamda)
%M=3;
%N=M;
%L=10;
%error=0;
%X = lamda.*[0 1 2 3 4; 5 6 7 8 9; 10 11 12 13 14; 15 16 17 18 19; 20
21 22 23 24];
%Y = lamda.*[0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4];
%Y = zeros(M,N);
x_loc=L*rand(M,N)-L/2;
y_loc=L*rand(M,N)-L/2;

%x_loc=L*X;
%y_loc=L*Y;

x_loc(1,1)=zeros(1,1);
y_loc(1,1)=zeros(1,1);
dx=-error+(2*error)*rand(M,N);
dy=-error+(2*error)*rand(M,N);
x_err=x_loc+dx;
y_err=y_loc+dy;

for i = 1:M
    for j = 1:N
distance(i,j)   =   sqrt((x_err(1,1)-x_err(i,j))^2   +   (y_err(1,1)-
y_err(i,j))^2);
midpointX(i,j) = (x_err(1,1)+x_err(i,j))/2;
```

```
midpointY(i,j) = (y_err(1,1)+y_err(i,j))/2;
    end
end
distance = distance
midpointX = midpointX
midpointY = midpointY
x=x_loc;
y=y_loc;
save ARRAY x y x_err y_err distance midpointX midpointY
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. BEAMFORMING

```
%  ---------------------------------------------------------------------
--------------------
% Filename: Random_MAIN.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
% error = position errors in meters (assumed uniformly distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% AF_mag = array factor
%  ---------------------------------------------------------------------
--------------------

%function[]=Random_MAIN()

clear

Na=10;     %input('Enter number of random array samples to be generated
:');
M=5;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
%disp('Number of elements along y-axis, N = M')
wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=50;           %input('Enter length of sensor cluster square area in
meters :');
error1= 0.0;   %input('Enter position error in meters :');
fail1=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error2= wavelength/2;   %input('Enter position error in meters :');
fail2=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error3= wavelength/2;   %input('Enter position error in meters :');
fail3=5.0;       %input('Enter number of elements failed (must be less
than MxN) :');
theta0=pi/4; %pi*rand(1,1)-pi/2;  %input('Enter of AOA elevation angle
in radians (-pi/2 < angle < pi/2) :');
theta1=pi/4;  %theta0-pi/10 %  pi*rand(1,1)-pi/2;  %input('Enter of AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
```

```
theta2=pi/4;  %theta0+pi/10  %pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
phi01=pi/4 %disp('Assume AOA azimuth angle = pi/4 radians (45 degrees)
');
phi02=pi/4
phi03=pi/4
phi1=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume AOA azimuth angle = pi/4
radians (45 degrees) ');
phi2=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume AOA azimuth angle = pi/4
radians (45 degrees) ');
fs=500;       %input('Enter sampling rate (>300) :');

AF_mag_sum_theta1=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta2=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta3=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_phi1=zeros(1,6284);
AF_mag_sum_phi2=zeros(1,6284);
AF_mag_sum_phi3=zeros(1,6284);
%load('Array');
%[x,x_err,y,y_err,distance,midpointX,midpointY]=Random_LOC_DOA(M,N,L,er
ror2);

   % figure(1), plot(x,y,'*'), grid on, hold on
for j=1:1:Na
    j
    [x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
    x(1,1)=0.0;
    x_err(1,1)=0.0;
    y(1,1)=0.0;
    y_err(1,1)=0.0;
    figure(1), plot(x,y,'*'), grid on, hold on

[mag_Wts1,ang_Wts1]=LMS(N,M,x,y,theta0,theta1,theta2,phi01,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail1]=Random_FAIL(N,M,mag_Wts1,fail1);
    [THETA1,
AF_mag_theta1]=AF_cal_theta(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
phi01);
    [PHI1,
AF_mag_phi1]=AF_cal_phi(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
theta0);
    AF_mag_sum_theta1=AF_mag_sum_theta1+AF_mag_theta1;
    AF_mag_sum_phi1=AF_mag_sum_phi1+AF_mag_phi1;


[mag_Wts2,ang_Wts2]=LMS(N,M,x,y,theta0,theta1,theta2,phi02,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail2]=Random_FAIL(N,M,mag_Wts2,fail2);
    [THETA2,
AF_mag_theta2]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,wave
length, phi02);
    [PHI2,
AF_mag_phi2]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,waveleng
th, theta0);
```

```
    AF_mag_sum_theta2=AF_mag_sum_theta2+AF_mag_theta2;
    AF_mag_sum_phi2=AF_mag_sum_phi2+AF_mag_phi2;


[mag_Wts3,ang_Wts3]=LMS(N,M,x,y,theta0,theta1,theta2,phi03,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail3]=Random_FAIL(N,M,mag_Wts3,fail3);
    [THETA3,
AF_mag_theta3]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,wave
length, phi03);
    [PHI3,
AF_mag_phi3]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,waveleng
th, theta0);
    AF_mag_sum_theta3=AF_mag_sum_theta3+AF_mag_theta3;
    AF_mag_sum_phi3=AF_mag_sum_phi3+AF_mag_phi3;


end
xlabel('x-direction in meters (m)'), ylabel('y-direction in meters
(m)')
title('50 square meter grid')
avg_AF_mag_theta1=AF_mag_sum_theta1;
avg_AF_mag_phi1=AF_mag_sum_phi1;

%avg_AF_mag_theta2=AF_mag_sum_theta2;
avg_AF_mag_phi2=AF_mag_sum_phi2;

%avg_AF_mag_theta3=AF_mag_sum_theta3;
avg_AF_mag_phi3=AF_mag_sum_phi3;

theta=(THETA1/pi)*180;
phi1=(PHI1/pi)*180;
phi2=(PHI2/pi)*180;
phi3=(PHI3/pi)*180;

theta_degrees=(theta0/pi)*180
phi_degrees1=(phi01/pi)*180
phi_degrees2=(phi02/pi)*180
phi_degrees3=(phi03/pi)*180

% 2D plot
% --------
figure (2), subplot(3,1,1)
plot(phi1,20*log10(avg_AF_mag_phi1),'r'), grid on
xlabel('Azimuth Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'),
xlim([0 360]),ylim([-10 30])
subplot(3,1,2)
plot(phi2,20*log10(avg_AF_mag_phi2),'b'), grid on
xlabel('Azimuth Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'),
xlim([0 360]),ylim([-10 30])
title('1 meter error');
subplot(3,1,3)
plot(phi3,20*log10(avg_AF_mag_phi3),'g'), grid on
xlabel('Azimuth Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'),
```

```
xlim([0 360]),ylim([-10 30])
title('1 meter error and 20 nodes fail')
plotedit on
zoom on
orient tall

% 2D Polar plot
% --------------
figure (3), polardb1(THETA1,20*log10(avg_AF_mag_theta1),-60,'m'), grid
on
orient tall
figure (4), polar(PHI1,20*log10(avg_AF_mag_phi1),'r'), grid on, hold on
polar(PHI2,20*log10(avg_AF_mag_phi2),'b'),
polar(PHI3,20*log10(avg_AF_mag_phi3),'g')
legend('No errors and no failures','\lambda/2 error in node location',
'\lambda/2 error in node location and 5 node failures')
title('Adaptive Beamforming - 157 MHz')
gtext('50 square meter grid, where \lambda = 1.9108 m')
orient tall

%PG_plot(THETA,avg_AF_mag_theta)
%PG_plot(PHI,avg_AF_mag_phi)
%PG_plot_polar(THETA,avg_AF_mag)
%load('AF_256_0deg_avg.mat','THETA','avg_AF_mag');
polardb1(THETA,20*log10(avg_AF_mag),-60,'m');
```

APPENDIX J. RANDOM_LOC

```
%  ---------------------------------------------------------------------
---------------------
% Filename: Random_LOC.m
%  ---------------------------------------------------------------------
---------------------
% Input parameters:
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% L = length of square planar area (in meters)
% error = position errors (assumed uniformly distributed between [-
posterror and +posterror])
%  ---------------------------------------------------------------------
---------------------
% Output:
% x = x-coordinates of sensor nodes (in meters)
% x_err = x-coordinates of sensor nodes with errors (in meters)
% y = y-coordinates of sensor nodes (in meters)
% y_err = y-coordinates of sensor nodes with errors (in meters)
%  ---------------------------------------------------------------------
---------------------

function[x,x_err,y,y_err] = Random_LOC(M,N,L,error)

x=L*rand(M,N);
y=L*rand(M,N);

dx=-error+(2*error)*rand(M,N);
dy=-error+(2*error)*rand(M,N);
x_err=x+dx;
y_err=y+dy;
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX K. 3D BEAMFORMING

```
%  ---------------------------------------------------------------------
--------------------
% Filename: Beam3D.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
% error = position errors in meters (assumed uniformly distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% AF_mag = array factor
%  ---------------------------------------------------------------------
--------------------

%function[]=Random_MAIN()

clear

Na=10;     %input('Enter number of random array samples to be generated
:');
M=5;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
%disp('Number of elements along y-axis, N = M')
wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=10;           %input('Enter length of sensor cluster square area in
meters :');
H=1000; % max array height
velocity = 10.0;
error1= 0.0;   %input('Enter position error in meters :');
fail1=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
theta0=-pi/4; %pi*rand(1,1)-pi/2;  %input('Enter of AOA elevation angle
in radians (-pi/2 < angle < pi/2) :');
theta1=0.0;
theta2=0.0;

phi01=pi/4; %disp('Assume AOA azimuth angle = pi/4 radians (45 degrees)
');
phi1=0.0; %2*pi*rand(1,1)-pi; %disp('Assume AOA azimuth angle = pi/4
radians (45 degrees) ');
```

```
phi2=0.0;
fs1=500;       %input('Enter sampling rate (>300) :');
AF_mag_sum_theta1=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_phi1=zeros(1,6284);
%load('Array3D');
```
% Establishes the initial node positions
```
[x,x_err,y,y_err,z,z_err]=Random_LOC_DOA3D(M,N,H,L,error1);
    %x_err(1,1)=0.0;
    %x=x_err;
    %y_err(1,1)=0.0;
    %y=y_err;
    %z=z_err;
tic
for j=1:1:Na
    j
    %[x,x_err,y,y_err,z,z_err]=Random_LOC_DOA3D(M,N,H,L,error1);
    %x(1,1)=0.0;
    %x_err(1,1)=0.0;
    %y(1,1)=0.0;
    %y_err(1,1)=0.0;
    %z_err(1,1)=0.0;

    figure(1), plot3(x,y, z,'*'), grid on, hold on

[mag_Wts1,ang_Wts1]=LMS3D(N,M,H,x,y,z,theta0,theta1,theta2,phi01,phi1,p
hi2,fs1,wavelength,f);
    [mag_Wts_fail1]=Random_FAIL(N,M,mag_Wts1,fail1);
    [THETA1,
AF_mag_theta1]=AF_cal_theta3D(N,M,x,y,z,mag_Wts_fail1,ang_Wts1,waveleng
th, phi01);
    [PHI1,
AF_mag_phi1]=AF_cal_phi3D(N,M,x,y,z,mag_Wts_fail1,ang_Wts1,wavelength,
theta0);
    AF_mag_sum_theta1=AF_mag_sum_theta1+AF_mag_theta1;
    AF_mag_sum_phi1=AF_mag_sum_phi1+AF_mag_phi1;

[x,x_err,y,y_err,z,z_err]=Random_LOC_DOA3Dvelocity(M,N,H,x,y,z,velocity
);

end
avg_AF_mag_theta1=AF_mag_sum_theta1;
avg_AF_mag_phi1=AF_mag_sum_phi1;
time_elapsed = toc

theta1=(THETA1/pi)*180;
phi1=(PHI1/pi)*180;

% 2D Polar plot
% --------------
%figure (2), polardb1(THETA1,20*log10(avg_AF_mag_theta1),-60,'m'), grid
on
%orient tall
figure (2), polar(THETA1,20*log10(avg_AF_mag_theta1),'r'), grid on,
hold on
```

```
axis auto

title(‘Adaptive Beamforming – 3D’)
%gtext(‘10 square meter grid’)
orient tall
figure (3), polar(PHI1,20*log10(avg_AF_mag_phi1),’r’), grid on, hold on

axis auto

title(‘Adaptive Beamforming – 3D’)
%gtext(‘25 square meter grid’)
orient tall
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX L. RANDOM_LOC_VELOCITY

```
%  ------------------------------------------------------------------------
--------------------
% Filename: Random_LOC3Dvelocity.m
% ------------------------------------------------------------------
% Input parameters:
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% L = length of square planar area (in meters)
% error = position errors (assumed uniformly distributed between [-
posterror and +posterror])
%  ------------------------------------------------------------------------
--------------------
% Output:
% x = x-coordinates of sensor nodes (in meters)
% x_err = x-coordinates of sensor nodes with errors (in meters)
% y = y-coordinates of sensor nodes (in meters)
% y_err = y-coordinates of sensor nodes with errors (in meters)
% distance = distance between the sensor nodes and the reference node
in
% meters
%  ------------------------------------------------------------------------
--------------------

function[x_loc,x_err,y_loc,y_err,z_loc,z_err,distance,midpointX,midpoin
tY, midpointZ] = Random_LOC(M,N,H,x,y,z,velocity)
%M=3;
%N=M;
%L=10;
%error=0;
%X = lamda.*[0 1 2 3 4; 5 6 7 8 9; 10 11 12 13 14; 15 16 17 18 19; 20
21 22 23 24];
%Y = lamda.*[0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4; 0 1 2 3 4];
%Y = zeros(M,N);
dx=-velocity+(2*velocity)*rand(M,N);
dy=-velocity+(2*velocity)*rand(M,N);
dz=-velocity+(2*velocity)*rand(M,N);
x_loc=x+dx; %+rand(M,N);
y_loc=y+dy; %+rand(M,N);
z_loc=z+dz; %+rand(M,N);

%x_loc=L*X;
%y_loc=L*Y;

x_loc(1,1)=zeros(1,1);
y_loc(1,1)=zeros(1,1);

x_err=x_loc;%+dx;
y_err=y_loc;%+dy;
z_err=z_loc;%+dz;

for i = 1:M
    for j = 1:N
```

```
distance(i,j)    =    sqrt((x_err(1,1)-x_err(i,j))^2    +    (y_err(1,1)-
y_err(i,j))^2 + (z_err(1,1)-z_err(i,j))^2);
midpointX(i,j) = (x_err(1,1)+x_err(i,j))/2;
midpointY(i,j) = (y_err(1,1)+y_err(i,j))/2;
midpointZ(i,j) = (z_err(1,1)+z_err(i,j))/2;
    end
end
distance = distance;
midpointX = midpointX;
midpointY = midpointY;
midpointZ = midpointZ;
x=x_loc;
y=y_loc;
z=z_loc;
%save  ARRAY3D  x  y  z  x_err  y_err  z_err  distance  midpointX  midpointY
midpointZ
```

APPENDIX M. BEAMSCAN

```
% ----------------------------------------------------------------
% Filename: Random_MAIN.m
%  ----------------------------------------------------------------
--------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
%  error  =  position  errors  in  meters  (assumed  uniformly  distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ----------------------------------------------------------------
--------------------
% Output:
% AF_mag = array factor
%  ----------------------------------------------------------------
--------------------

%function[]=Random_MAIN()

clear

Na=10;     %input('Enter number of random array samples to be generated
:');
M=5;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
%disp('Number of elements along y-axis, N = M')
wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=100*wavelength;         %input('Enter length of sensor cluster square
area in meters :');
error1= 0.0;   %input('Enter position error in meters :');
fail1=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
error2= 0.0;   %input('Enter position error in meters :');
fail2=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
error3= 0.0;   %input('Enter position error in meters :');
fail3=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
error4= 0.0;   %input('Enter position error in meters :');
fail4=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
error5= 0.0;   %input('Enter position error in meters :');
fail5=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
```

```
error6= 0.0;   %input('Enter position error in meters :');
fail6=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');
error7= 0.0;   %input('Enter position error in meters :');
fail7=0.0;      %input('Enter number of elements failed (must be less
than MxN) :');


theta0=pi/10; %pi*rand(1,1)-pi/2;  %input('Enter of AOA elevation angle
in radians (-pi/2 < angle < pi/2) :');
theta1=pi/10; %theta0-pi/10 % pi*rand(1,1)-pi/2; %input('Enter of AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
theta2=pi/10;  %theta0+pi/10  %pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
theta3=pi/10;
theta4=pi/10;
theta5=pi/10;
theta6=pi/10;
phi01=pi/3; %disp('Assume AOA azimuth angle = pi/4 radians (45 degrees)
');
phi02=pi/4;
phi03=pi/6;
phi04=pi/7;
phi05=pi/8;
phi06=pi/9;
phi07=pi/10;
phi1=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume  AOA  azimuth  angle  =  pi/4
radians (45 degrees) ');
phi2=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume  AOA  azimuth  angle  =  pi/4
radians (45 degrees) ');
fs=500;       %input('Enter sampling rate (>300) :');

AF_mag_sum_theta1=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta2=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta3=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta4=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta5=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta6=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function
AF_mag_sum_theta7=zeros(1,3142);         % Do  not  change  this,  it  will
affect the AF_cal function

AF_mag_sum_phi1=zeros(1,6284);
AF_mag_sum_phi2=zeros(1,6284);
AF_mag_sum_phi3=zeros(1,6284);
AF_mag_sum_phi4=zeros(1,6284);
AF_mag_sum_phi5=zeros(1,6284);
AF_mag_sum_phi6=zeros(1,6284);
AF_mag_sum_phi7=zeros(1,6284);
%load('Array');
%[x,x_err,y,y_err,distance,midpointX,midpointY]=Random_LOC_DOA(M,N,L,er
ror2);
```

```
   % figure(1), plot(x,y,'*'), grid on, hold on
for j=1:1:Na
    j
    [x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
    x(1,1)=0.0;
    x_err(1,1)=0.0;
    y(1,1)=0.0;
    y_err(1,1)=0.0;
    figure(1), plot(x,y,'*'), grid on, hold on
    tic

[mag_Wts1,ang_Wts1]=LMS(N,M,x,y,theta0,theta1,theta2,phi01,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail1]=Random_FAIL(N,M,mag_Wts1,fail1);
    [THETA1,
AF_mag_theta1]=AF_cal_theta(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
phi01);
    [PHI1,
AF_mag_phi1]=AF_cal_phi(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
theta0);
    AF_mag_sum_theta1=AF_mag_sum_theta1+AF_mag_theta1;
    AF_mag_sum_phi1=AF_mag_sum_phi1+AF_mag_phi1;
    time_elapsed1 = toc

    tic

[mag_Wts2,ang_Wts2]=LMS(N,M,x,y,theta0,theta1,theta2,phi02,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail2]=Random_FAIL(N,M,mag_Wts2,fail2);
    [THETA2,
AF_mag_theta2]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,wave
length, phi02);
    [PHI2,
AF_mag_phi2]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,waveleng
th, theta0);
    AF_mag_sum_theta2=AF_mag_sum_theta2+AF_mag_theta2;
    AF_mag_sum_phi2=AF_mag_sum_phi2+AF_mag_phi2;
    time_elapsed2 = toc

    tic

[mag_Wts3,ang_Wts3]=LMS(N,M,x,y,theta0,theta1,theta2,phi03,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail3]=Random_FAIL(N,M,mag_Wts3,fail3);
    [THETA3,
AF_mag_theta3]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,wave
length, phi03);
    [PHI3,
AF_mag_phi3]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,waveleng
th, theta0);
    AF_mag_sum_theta3=AF_mag_sum_theta3+AF_mag_theta3;
    AF_mag_sum_phi3=AF_mag_sum_phi3+AF_mag_phi3;
    time_elapsed3 = toc

    tic
```

```
[mag_Wts4,ang_Wts4]=LMS(N,M,x,y,theta0,theta1,theta2,phi04,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail4]=Random_FAIL(N,M,mag_Wts4,fail4);
    [THETA4,
AF_mag_theta4]=AF_cal_theta(N,M,x,y,mag_Wts_fail4,ang_Wts4,wavelength,
phi04);
    [PHI4,
AF_mag_phi4]=AF_cal_phi(N,M,x,y,mag_Wts_fail4,ang_Wts4,wavelength,
theta0);
    AF_mag_sum_theta4=AF_mag_sum_theta4+AF_mag_theta4;
    AF_mag_sum_phi4=AF_mag_sum_phi4+AF_mag_phi4;
    time_elapsed4 = toc

    tic

[mag_Wts5,ang_Wts5]=LMS(N,M,x,y,theta0,theta1,theta2,phi05,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail5]=Random_FAIL(N,M,mag_Wts5,fail5);
    [THETA5,
AF_mag_theta5]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail5,ang_Wts5,wave
length, phi05);
    [PHI5,
AF_mag_phi5]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail5,ang_Wts5,waveleng
th, theta0);
    AF_mag_sum_theta5=AF_mag_sum_theta5+AF_mag_theta5;
    AF_mag_sum_phi5=AF_mag_sum_phi5+AF_mag_phi5;
    time_elapsed5 = toc

    tic

[mag_Wts6,ang_Wts6]=LMS(N,M,x,y,theta0,theta1,theta2,phi06,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail6]=Random_FAIL(N,M,mag_Wts6,fail6);
    [THETA6,
AF_mag_theta6]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail6,ang_Wts6,wave
length, phi06);
    [PHI6,
AF_mag_phi6]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail6,ang_Wts6,waveleng
th, theta0);
    AF_mag_sum_theta6=AF_mag_sum_theta6+AF_mag_theta6;
    AF_mag_sum_phi6=AF_mag_sum_phi6+AF_mag_phi6;
    time_elapsed6 = toc

    tic

[mag_Wts7,ang_Wts7]=LMS(N,M,x,y,theta0,theta1,theta2,phi07,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail7]=Random_FAIL(N,M,mag_Wts7,fail7);
    [THETA7,
AF_mag_theta7]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail7,ang_Wts7,wave
length, phi07);
    [PHI7,
AF_mag_phi7]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail7,ang_Wts7,waveleng
th, theta0);
    AF_mag_sum_theta7=AF_mag_sum_theta7+AF_mag_theta7;
    AF_mag_sum_phi7=AF_mag_sum_phi7+AF_mag_phi7;
```

```
    time_elapsed7 = toc
end
xlabel('x-direction in meters (m)'), ylabel('y-direction in meters
(m)')
title('100\lambda square meter grid')
avg_AF_mag_theta1=AF_mag_sum_theta1;
avg_AF_mag_phi1=AF_mag_sum_phi1;

%avg_AF_mag_theta2=AF_mag_sum_theta2;
avg_AF_mag_phi2=AF_mag_sum_phi2;

%avg_AF_mag_theta3=AF_mag_sum_theta3;
avg_AF_mag_phi3=AF_mag_sum_phi3;
avg_AF_mag_phi4=AF_mag_sum_phi4;
avg_AF_mag_phi5=AF_mag_sum_phi5;
avg_AF_mag_phi6=AF_mag_sum_phi6;
avg_AF_mag_phi7=AF_mag_sum_phi7;
%fname = input('Save average array factor data for elevation angle as
(with .mat extension) :  ', 's');
%save(fname,'THETA','avg_AF_mag_theta')
%fname = input('Save average array factor data for azimuth as (with
.mat extension) :  ', 's');
%save(fname,'PHI','avg_AF_mag_phi')

theta=(THETA1/pi)*180;
phi1=(PHI1/pi)*180;
phi2=(PHI2/pi)*180;
phi3=(PHI3/pi)*180;
phi4=(PHI4/pi)*180;
phi5=(PHI5/pi)*180;
phi6=(PHI6/pi)*180;
phi7=(PHI7/pi)*180;

theta_degrees=(theta0/pi)*180
phi_degrees1=(phi01/pi)*180
phi_degrees2=(phi02/pi)*180
phi_degrees3=(phi03/pi)*180
phi_degrees4=(phi04/pi)*180
phi_degrees5=(phi05/pi)*180
phi_degrees6=(phi06/pi)*180
phi_degrees7=(phi07/pi)*180

% 2D plot
% --------

figure (2), polar(PHI1,20*log10(avg_AF_mag_phi1),'r'), grid on, hold on
polar(PHI2,20*log10(avg_AF_mag_phi2),'b'),
polar(PHI3,20*log10(avg_AF_mag_phi3),'g'),
polar(PHI4,20*log10(avg_AF_mag_phi4),'k')
polar(PHI5,20*log10(avg_AF_mag_phi5),'m'),
polar(PHI6,20*log10(avg_AF_mag_phi6),'y'),
polar(PHI7,20*log10(avg_AF_mag_phi7),'c')
legend('\lambda/3  angle-of-arrival',  '\lambda/4  angle-of-arrival',
'\lambda/6  angle-of-arrival','\lambda/7  angle-of-arrival',  '\lambda/8
angle-of-arrival',  '\lambda/9  angle-of-arrival','\lambda/10  angle-of-
arrival')
title('Adaptive Beamforming – 157 MHz')
```

```
gtext('100\lambda square meter grid, where \lambda = 1.9108 m')
orient tall

%PG_plot(THETA,avg_AF_mag_theta)
%PG_plot(PHI,avg_AF_mag_phi)
%PG_plot_polar(THETA,avg_AF_mag)
%load('AF_256_0deg_avg.mat','THETA','avg_AF_mag');
polardb1(THETA,20*log10(avg_AF_mag),-60,'m');
```

APPENDIX N. LEAST MEAN SQUARE (LMS)

```
%  ---------------------------------------------------------------------
--------------------
% Filename: LMS.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% N = number of elements along y-axis (N must be equal to M)
% M = number of elements along x-axis (M must be equal to N)
% x = x-coordinates of sensor nodes (in meters)
% y = y-coordinates of sensor nodes (in meters)
% theta0 = AOA theta angle (assume AOA phi angle = 45 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% mag_Wts = magnitude of the weights
% ang_Wts = angle of the weights (in radians)
%  ---------------------------------------------------------------------
--------------------

function[mag_Wts,ang_Wts]                                                =
LMS(N,M,x,y,theta0,theta1,theta2,phi0,phi1,phi2,fs,wavelength,f)

% ------------------
% Frequency settings
% ------------------
c=3e8;
mfreq0=c/wavelength;            % set freq of desired modulating signal
= 100 Hz
mfreq1=mfreq0;             % set freq of interference modulating signal
f1
mfreq2=mfreq0;             % set freq of interference modulating signal
f2
fsamp=2*mfreq0;
lamdad=wavelength;    % wavelength of design freq of the array
lamda0=wavelength;        %  wavelength  of  carrier  freq  of  desired
modulating signal
lamda1=wavelength;      %  wavelength  of  carrier  freq  of  interference
modulating signal f1
lamda2=wavelength;      %  wavelength  of  carrier  freq  of  interference
modulating signal f2

% ---------------------------------
% Steering vector for desired signal
% ---------------------------------
% theta0=0;                                          % main beam arriving
elevation angle for desired signal
%phi0=pi/4;                                                 % main beam
arriving azimuth angle for desired signal
for n0=1:1:N
    for m0=1:1:M
```

```
Array_vector0(m0,n0)=exp(j*(2*pi/lamda0)*(x(m0,n0)*sin(theta0)*cos(phi0
)+y(m0,n0)*sin(theta0)*sin(phi0)));
    end
end

% ----------------------------------------
% Steering vector for interference signal f1
% ----------------------------------------
theta1=pi/3;                                          %main beam arriving
elevation angle for interference signal f1
%phi1=pi/4;                                                  % main beam
arriving azimuth angle for interference signal f1
for n1=1:1:N
    for m1=1:1:M

Array_vector1(m1,n1)=exp(j*(2*pi/lamda1)*(x(m1,n1)*sin(theta1)*cos(phi1
)+y(m1,n1)*sin(theta1)*sin(phi1)));
    end
end
% ----------------------------------------
% Steering vector for interference signal f2
% ----------------------------------------
theta2=-pi/3;                                        % main beam arriving
elevation angle for interference signal f2
%phi2=pi/4;                                                  % main beam
arriving azimuth angle for interference signal f2
for n2=1:1:N
    for m2=1:1:M

Array_vector2(m2,n2)=exp(j*(2*pi/lamda2)*(x(m2,n2)*sin(theta2)*cos(phi2
)+y(m2,n2)*sin(theta2)*sin(phi2)));
    end
end

% -------------
% LMS algorithm
% -------------
% fs=500;                                            % sampling rate
w(N,M,1)=zeros;                                           % set initial
weights to zero
conv_factor=0.001;                                  % convergent factor
for LMS algorithm
phase0=2*pi*rand(1,fs);                              % generate uniform
random phase for desired signal
phase1=2*pi*rand(1,fs);                              % generate uniform
random phase for interference signal f1
phase2=2*pi*rand(1,fs);                              % generate uniform
random phase for interference signal f2
sigma=1e-6;  %0.1 * set the noise power
for t=1:1:fs;
    noise(:,:,t)=sigma*randn(M,N);
    r(t)=cos(2*pi*mfreq0*t/fsamp+phase0(t));                          %
reference signal
    s(t)=cos(2*pi*mfreq0*t/fsamp+phase0(t));                          %
Desired modulating signal
```

```
    f1(t)=cos(2*pi*mfreq1*t/fsamp+phase1(t));                          %
interference modulating signal f1
    f2(t)=cos(2*pi*mfreq2*t/fsamp+phase2(t));                          %
interference modulating signal f2
    X(:,:,t)=s(t)*Array_vector0      +      f1(t)*Array_vector1      +
f2(t)*Array_vector2 + noise(:,:,t);
    e(t)=conj(r(t))-sum(sum((conj(X(:,:,t))).*w(:,:,t)));              %
calculate the conjugate of the error
    w(:,:,t+1)=w(:,:,t)  +  conv_factor*X(:,:,t)*e(t);                 %
calculate new weights for next iteration
    eig_value(:,t)=eig(X(:,:,t)*(conj(X(:,:,t)).'));                   %
calculate eigenvalue of xx* during each sample
    max_eigvalue(t)=max(eig_value(:,t));                               %
calculate maximum eigenvalue of xx* during each sample
end

% -------------------------
% Graphical results display
% -------------------------
%square_error=abs((e(1:length(e)).*e(1:length(e))));
% plot(1:length(e),square_error)
%Mean_sq_error=sum(square_error)/fs
% Title('Convergence of the LMS algorithm')
% xlabel('No. of samples'), ylabel('Magnitude(Error^2)'), xlim([0 fs]),
grid on
% -------------------------
% Numerical results display
% -------------------------
%Max_convergence_factor = 2/max(max_eigvalue);
mag_Wts=abs(conj(w(:,:,t)));
ang_Wts=angle(conj(w(:,:,t)));
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX O. LEAST MEAN SQUARE 3D

```
%  ---------------------------------------------------------------------
--------------------
% Filename: LMS3D.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% N = number of elements along y-axis (N must be equal to M)
% M = number of elements along x-axis (M must be equal to N)
% x = x-coordinates of sensor nodes (in meters)
% y = y-coordinates of sensor nodes (in meters)
% theta0 = AOA theta angle (assume AOA phi angle = 45 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% mag_Wts = magnitude of the weights
% ang_Wts = angle of the weights (in radians)
%  ---------------------------------------------------------------------
--------------------

function[mag_Wts,ang_Wts]                                                =
LMS(N,M,H,x,y,z,theta0,theta1,theta2,phi0,phi1,phi2,fs,wavelength,f)

% ------------------
% Frequency settings
% ------------------
c=3e8;
mfreq0=c/wavelength;            % set freq of desired modulating signal
= 100 Hz
mfreq1=mfreq0;             % set freq of interference modulating signal
f1
mfreq2=mfreq0;             % set freq of interference modulating signal
f2
fsamp=2*mfreq0;
lamdad=wavelength;    % wavelength of design freq of the array
lamda0=wavelength;        %  wavelength  of  carrier  freq  of  desired
modulating signal
lamda1=wavelength;      %  wavelength  of  carrier  freq  of  interference
modulating signal f1
lamda2=wavelength;      %  wavelength  of  carrier  freq  of  interference
modulating signal f2

% ---------------------------------
% Steering vector for desired signal
% ---------------------------------
% theta0=0;                                          % main beam arriving
elevation angle for desired signal
%phi0=pi/4;                                                 % main beam
arriving azimuth angle for desired signal
for n0=1:1:N
    for m0=1:1:M
```

```
Array_vector0(m0,n0)=exp(j*(2*pi/lamda0)*(x(m0,n0)*sin(theta0)*cos(phi0
)+y(m0,n0)*sin(theta0)*sin(phi0)+z(m0,n0)));
    end
end

% ----------------------------------------
% Steering vector for interference signal f1
% ----------------------------------------
theta1=pi/3;                                          %main beam arriving
elevation angle for interference signal f1
%phi1=pi/4;                                                  % main beam
arriving azimuth angle for interference signal f1
for n1=1:1:N
    for m1=1:1:M

Array_vector1(m1,n1)=exp(j*(2*pi/lamda1)*(x(m1,n1)*sin(theta1)*cos(phi1
)+y(m1,n1)*sin(theta1)*sin(phi1)+z(m1,n1)));
    end
end
% ----------------------------------------
% Steering vector for interference signal f2
% ----------------------------------------
theta2=-pi/3;                                        % main beam arriving
elevation angle for interference signal f2
%phi2=pi/4;                                                  % main beam
arriving azimuth angle for interference signal f2
for n2=1:1:N
    for m2=1:1:M

Array_vector2(m2,n2)=exp(j*(2*pi/lamda2)*(x(m2,n2)*sin(theta2)*cos(phi2
)+y(m2,n2)*sin(theta2)*sin(phi2)+z(m2,n2)));
    end
end

% -------------
% LMS algorithm
% -------------
% fs=500;                                            % sampling rate
w(N,M,1)=zeros;                                           % set initial
weights to zero
conv_factor=0.001;                                    % convergent factor
for LMS algorithm
phase0=2*pi*rand(1,fs);                                % generate uniform
random phase for desired signal
phase1=2*pi*rand(1,fs);                                % generate uniform
random phase for interference signal f1
phase2=2*pi*rand(1,fs);                                % generate uniform
random phase for interference signal f2
sigma=1e-6;  %0.1 * set the noise power
for t=1:1:fs;
    noise(:,:,t)=sigma*randn(M,N);
    r(t)=cos(2*pi*mfreq0*t/fsamp+phase0(t));                              %
reference signal
    s(t)=cos(2*pi*mfreq0*t/fsamp+phase0(t));                              %
Desired modulating signal
```

```
    f1(t)=cos(2*pi*mfreq1*t/fsamp+phase1(t));                          %
interference modulating signal f1
    f2(t)=cos(2*pi*mfreq2*t/fsamp+phase2(t));                          %
interference modulating signal f2
    X(:,:,t)=s(t)*Array_vector0      +      f1(t)*Array_vector1      +
f2(t)*Array_vector2 + noise(:,:,t);
    e(t)=conj(r(t))-sum(sum((conj(X(:,:,t))).*w(:,:,t)));              %
calculate the conjugate of the error
    w(:,:,t+1)=w(:,:,t)  +  conv_factor*X(:,:,t)*e(t);                 %
calculate new weights for next iteration
    eig_value(:,t)=eig(X(:,:,t)*(conj(X(:,:,t)).'));                   %
calculate eigenvalue of xx* during each sample
    max_eigvalue(t)=max(eig_value(:,t));                               %
calculate maximum eigenvalue of xx* during each sample
end

% -------------------------
% Graphical results display
% -------------------------
%square_error=abs((e(1:length(e)).*e(1:length(e))));
% plot(1:length(e),square_error)
%Mean_sq_error=sum(square_error)/fs
% Title('Convergence of the LMS algorithm')
% xlabel('No. of samples'), ylabel('Magnitude(Error^2)'), xlim([0 fs]),
grid on
% -------------------------
% Numerical results display
% -------------------------
%Max_convergence_factor = 2/max(max_eigvalue);
       mag_Wts=abs(conj(w(:,:,t)));ang_Wts=angle(conj(w(:,:,t)));
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX P. BEAMWIDTH

```
%  ---------------------------------------------------------------------
--------------------
% Filename: Beamwidth.m
%  ---------------------------------------------------------------------
--------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
% error = position errors in meters (assumed uniformly distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ---------------------------------------------------------------------
--------------------
% Output:
% AF_mag = array factor
%  ---------------------------------------------------------------------
--------------------

%function[]=Random_MAIN()

clear

Na=20;     %input('Enter number of random array samples to be generated
:');
M=5;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
%disp('Number of elements along y-axis, N = M')
wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=5*wavelength;          %input('Enter length of sensor cluster square
area in meters :');
error1= 0.0;   %input('Enter position error in meters :');
fail1=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error2= 0.0; %wavelength/2;    %input('Enter position error in meters
:');
fail2=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error3= 0.0; %wavelength/2;    %input('Enter position error in meters
:');
fail3=0.0; %5.0;      %input('Enter number of elements failed (must be
less than MxN) :');
theta0=0.0; %pi*rand(1,1)-pi/2;   %input('Enter of AOA elevation angle
in radians (-pi/2 < angle < pi/2) :');
```

```
theta1=0.0;  %theta0-pi/10  %  pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
theta2=0.0;  %theta0+pi/10  %pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
theta3=0.0;
phi01=pi/4; %disp('Assume AOA azimuth angle = pi/4 radians (45 degrees)
');
phi02=pi/4;
phi03=pi/4;
phi1=pi/2; %2*pi*rand(1,1)-pi; %disp('Assume AOA azimuth angle = pi/4
radians (45 degrees) ');
phi2=pi/2; %2*pi*rand(1,1)-pi; %disp('Assume AOA azimuth angle = pi/4
radians (45 degrees) ');
phi3=pi/2;
fs1=50;       %input('Enter sampling rate (>300) :');
fs2=100;
fs3=500;
AF_mag_sum_theta1=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta2=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta3=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_phi1=zeros(1,6284);
AF_mag_sum_phi2=zeros(1,6284);
AF_mag_sum_phi3=zeros(1,6284);
%load('Array');
[x,x_err,y,y_err,distance,midpointX,midpointY]=Random_LOC_DOA(M,N,L,err
or2);
    %x=x_err;
    %y=y_err;
    figure(1), plot(x,y,'*'), grid on, hold on
for j=1:1:Na
    j
    %[x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
    %x(1,1)=0.0;
    %x_err(1,1)=0.0;
    %y(1,1)=0.0;
    %y_err(1,1)=0.0;


[mag_Wts1,ang_Wts1]=LMS(N,M,x,y,theta0,theta1,theta2,phi01,phi1,phi2,fs
1,wavelength,f);
    [mag_Wts_fail1]=Random_FAIL(N,M,mag_Wts1,fail1);
    [THETA1,
AF_mag_theta1]=AF_cal_theta(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
phi01);
    [PHI1,
AF_mag_phi1]=AF_cal_phi(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
theta0);
    AF_mag_sum_theta1=AF_mag_sum_theta1+AF_mag_theta1;
    AF_mag_sum_phi1=AF_mag_sum_phi1+AF_mag_phi1;
    %end
avg_AF_mag_theta1=AF_mag_sum_theta1;
avg_AF_mag_phi1=AF_mag_sum_phi1;

%for j=1:1:Na
```

```
    %[x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
    %x(1,1)=0.0;
    %x_err(1,1)=0.0;
    %y(1,1)=0.0;
    %y_err(1,1)=0.0;

[mag_Wts2,ang_Wts2]=LMS(N,M,x,y,theta0,theta1,theta2,phi02,phi1,phi2,fs
2,wavelength,f);
    [mag_Wts_fail2]=Random_FAIL(N,M,mag_Wts2,fail2);
    [THETA2,
AF_mag_theta2]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,wave
length, phi02);
    [PHI2,
AF_mag_phi2]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,waveleng
th, theta0);
    AF_mag_sum_theta2=AF_mag_sum_theta2+AF_mag_theta2;
    AF_mag_sum_phi2=AF_mag_sum_phi2+AF_mag_phi2;
    %end
avg_AF_mag_theta2=AF_mag_sum_theta2;
avg_AF_mag_phi2=AF_mag_sum_phi2;

%for j=1:1:Na

    %[x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
    %x(1,1)=0.0;
    %x_err(1,1)=0.0;
    %y(1,1)=0.0;
    %y_err(1,1)=0.0;

[mag_Wts3,ang_Wts3]=LMS(N,M,x,y,theta0,theta1,theta2,phi03,phi1,phi2,fs
3,wavelength,f);
    [mag_Wts_fail3]=Random_FAIL(N,M,mag_Wts3,fail3);
    [THETA3,
AF_mag_theta3]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,wave
length, phi03);
    [PHI3,
AF_mag_phi3]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,waveleng
th, theta0);
    AF_mag_sum_theta3=AF_mag_sum_theta3+AF_mag_theta3;
    AF_mag_sum_phi3=AF_mag_sum_phi3+AF_mag_phi3;
end
avg_AF_mag_theta3=AF_mag_sum_theta3;
avg_AF_mag_phi3=AF_mag_sum_phi3;
theta1=(THETA1/pi)*180;
theta2=(THETA2/pi)*180;
theta3=(THETA3/pi)*180;
phi1=(PHI1/pi)*180;
phi2=(PHI2/pi)*180;
phi3=(PHI3/pi)*180;
% 2D plot
% --------
figure (2), %subplot(3,1,1)
plot(theta1,20*log10(avg_AF_mag_theta1),'r'), hold on, grid on
%xlabel('Elevation Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'),
%xlim([-40 40]),ylim([-20 25])
```

```
%title('50 samples')
%subplot(3,1,2)
plot(theta2,20*log10(avg_AF_mag_theta2),'b'), grid on
%xlabel('Elevation Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'),
%xlim([-40 40]),ylim([-20 25])
%title('100 samples');
%subplot(3,1,3)
plot(theta3,20*log10(avg_AF_mag_theta3),'g'), grid on
xlabel('Elevation Angle-of-Arrival (degrees)'), ylabel('Average power
gain (dB)'), xlim([-40 40]),ylim([0 40])
title('25 Element Sensor Field (20 iterations)')
legend('50 samples','100 samples', '500 samples')
plotedit on
zoom on
orient tall

% 2D Polar plot
% --------------
%figure (2), polardb1(THETA1,20*log10(avg_AF_mag_theta1),-60,'m'), grid
on
%orient tall
figure (3), polar(THETA1,20*log10(avg_AF_mag_theta1),'r'), grid on,
hold on
polar(THETA2,20*log10(avg_AF_mag_theta2),'b'),
polar(THETA3,20*log10(avg_AF_mag_theta3),'g')
axis auto
legend('50 samples','100 samples', '500 samples')
title('Adaptive Beamforming - 157 MHz')
gtext('5\lambda square meter grid, where \lambda = 1.9108 m')
 orient tall% ------------------------------------------------------------
                    --------------------------------


% Filename: Node location errors
%  ------------------------------------------------------------------------
---------------------
% Input parameters:
% Na = number of random arrays to be generated
% M = number of elements along x-axis (M must be equal to N)
% N = number of elements along y-axis (N must be equal to M)
% wavelength = design wavelength of array (in meters)
% L = length of square planar (in meters)
%  error = position errors in meters (assumed uniformly distributed
between [-posterror and +posterror])
% fail = number of elements failure
% theta0 = AOA theta angle (assume AOA phi angle = pi/4 or 90 degrees)
% fs = sampling rate for the LMS algorithm
%  ------------------------------------------------------------------------
---------------------
% Output:
% AF_mag = array factor
%  ------------------------------------------------------------------------
---------------------

%function[]=Random_MAIN()
```

```
clear

Na=10;     %input('Enter number of random array samples to be generated
:');
M=5;      %input('Enter number of elements along x-axis, M (must be >1)
:');
N=M;
c=3e8;
f=157e6;
%disp('Number of elements along y-axis, N = M')
wavelength=c/f;    %input('Enter design wavelength of array in meters
:');
L=50*wavelength;          %input('Enter length of sensor cluster square
area in meters :');
error1= 0.0;   %input('Enter position error in meters :');
fail1=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error2= wavelength/2;   %input('Enter position error in meters :');
fail2=0.0;       %input('Enter number of elements failed (must be less
than MxN) :');
error3= wavelength/2;   %input('Enter position error in meters :');
fail3=5.0;       %input('Enter number of elements failed (must be less
than MxN) :');
theta0=pi/4; %pi*rand(1,1)-pi/2;  %input('Enter of AOA elevation angle
in radians (-pi/2 < angle < pi/2) :');
theta1=pi/4;  %theta0-pi/10 %  pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
theta2=pi/4;  %theta0+pi/10  %pi*rand(1,1)-pi/2;  %input('Enter  of  AOA
elevation angle in radians (-pi/2 < angle < pi/2) :');
phi01=pi/4 %disp('Assume AOA azimuth angle = pi/4 radians (45 degrees)
');
phi02=pi/4
phi03=pi/4
phi1=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume  AOA  azimuth  angle  =  pi/4
radians (45 degrees) ');
phi2=pi/10 %2*pi*rand(1,1)-pi; %disp('Assume  AOA  azimuth  angle  =  pi/4
radians (45 degrees) ');
fs=500;       %input('Enter sampling rate (>300) :');

AF_mag_sum_theta1=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta2=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_theta3=zeros(1,3142);         % Do not change this, it will
affect the AF_cal function
AF_mag_sum_phi1=zeros(1,6284);
AF_mag_sum_phi2=zeros(1,6284);
AF_mag_sum_phi3=zeros(1,6284);
%load('Array');
%[x,x_err,y,y_err,distance,midpointX,midpointY]=Random_LOC_DOA(M,N,L,er
ror2);

   % figure(1), plot(x,y,'*'), grid on, hold on
for j=1:1:Na
    j
    [x,x_err,y,y_err]=Random_LOC(M,N,L,error2);
```

```
    x(1,1)=0.0;
    x_err(1,1)=0.0;
    y(1,1)=0.0;
    y_err(1,1)=0.0;
    figure(1), plot(x,y,'*'), grid on, hold on

[mag_Wts1,ang_Wts1]=LMS(N,M,x,y,theta0,theta1,theta2,phi01,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail1]=Random_FAIL(N,M,mag_Wts1,fail1);
    [THETA1,
AF_mag_theta1]=AF_cal_theta(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
phi01);
    [PHI1,
AF_mag_phi1]=AF_cal_phi(N,M,x,y,mag_Wts_fail1,ang_Wts1,wavelength,
theta0);
    AF_mag_sum_theta1=AF_mag_sum_theta1+AF_mag_theta1;
    AF_mag_sum_phi1=AF_mag_sum_phi1+AF_mag_phi1;


[mag_Wts2,ang_Wts2]=LMS(N,M,x,y,theta0,theta1,theta2,phi02,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail2]=Random_FAIL(N,M,mag_Wts2,fail2);
    [THETA2,
AF_mag_theta2]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,wave
length, phi02);
    [PHI2,
AF_mag_phi2]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail2,ang_Wts2,waveleng
th, theta0);
    AF_mag_sum_theta2=AF_mag_sum_theta2+AF_mag_theta2;
    AF_mag_sum_phi2=AF_mag_sum_phi2+AF_mag_phi2;


[mag_Wts3,ang_Wts3]=LMS(N,M,x,y,theta0,theta1,theta2,phi03,phi1,phi2,fs
,wavelength,f);
    [mag_Wts_fail3]=Random_FAIL(N,M,mag_Wts3,fail3);
    [THETA3,
AF_mag_theta3]=AF_cal_theta(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,wave
length, phi03);
    [PHI3,
AF_mag_phi3]=AF_cal_phi(N,M,x_err,y_err,mag_Wts_fail3,ang_Wts3,waveleng
th, theta0);
    AF_mag_sum_theta3=AF_mag_sum_theta3+AF_mag_theta3;
    AF_mag_sum_phi3=AF_mag_sum_phi3+AF_mag_phi3;


end
xlabel('x-direction in meters (m)'), ylabel('y-direction in meters
(m)')
title('50\lambda square meter grid')
avg_AF_mag_theta1=AF_mag_sum_theta1;
avg_AF_mag_phi1=AF_mag_sum_phi1;

%avg_AF_mag_theta2=AF_mag_sum_theta2;
avg_AF_mag_phi2=AF_mag_sum_phi2;

%avg_AF_mag_theta3=AF_mag_sum_theta3;
avg_AF_mag_phi3=AF_mag_sum_phi3;
```

```
%fname = input('Save average array factor data for elevation angle as
(with .mat extension) :  ', 's');
%save(fname,'THETA','avg_AF_mag_theta')
%fname = input('Save average array factor data for azimuth as (with
.mat extension) :  ', 's');
%save(fname,'PHI','avg_AF_mag_phi')

theta=(THETA1/pi)*180;
phi1=(PHI1/pi)*180;
phi2=(PHI2/pi)*180;
phi3=(PHI3/pi)*180;

theta_degrees=(theta0/pi)*180
phi_degrees1=(phi01/pi)*180
phi_degrees2=(phi02/pi)*180
phi_degrees3=(phi03/pi)*180

% 2D plot
% --------
figure (2), subplot(3,1,1)
plot(phi1,20*log10(avg_AF_mag_phi1),'r'), grid on
xlabel('Azimuth  Angle-of-Arrival  (degrees)'),  ylabel('Average  power
gain (dB)'), xlim([0 360]),ylim([-10 30])
subplot(3,1,2)
plot(phi2,20*log10(avg_AF_mag_phi2),'b'), grid on
xlabel('Azimuth  Angle-of-Arrival  (degrees)'),  ylabel('Average  power
gain (dB)'), xlim([0 360]),ylim([-10 30])
title('1 meter error');
subplot(3,1,3)
plot(phi3,20*log10(avg_AF_mag_phi3),'g'), grid on
xlabel('Azimuth  Angle-of-Arrival  (degrees)'),  ylabel('Average  power
gain (dB)'), xlim([0 360]),ylim([-10 30])
title('1 meter error and 20 nodes fail')
plotedit on
zoom on
orient tall

% 2D Polar plot
% --------------
figure  (3),  polardb1(THETA1,20*log10(avg_AF_mag_theta1),-60,'m'),  grid
on
orient tall
figure (4), polar(PHI1,20*log10(avg_AF_mag_phi1),'r'), grid on, hold on
polar(PHI2,20*log10(avg_AF_mag_phi2),'b'),
polar(PHI3,20*log10(avg_AF_mag_phi3),'g')
legend('No errors and no failures','\lambda/2 error in node location',
'\lambda/2 error in node location and 5 node failures')
title('Adaptive Beamforming - 157 MHz')
gtext('50\lambda square meter grid, where \lambda = 1.9108 m')
orient tall

%PG_plot(THETA,avg_AF_mag_theta)
%PG_plot(PHI,avg_AF_mag_phi)
%PG_plot_polar(THETA,avg_AF_mag)
%load('AF_256_0deg_avg.mat','THETA','avg_AF_mag');
polardb1(THETA,20*log10(avg_AF_mag),-60,'m');
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX Q. CRAMÉR-RAO BOUND

Using the example taken from [71] the Cramér-Rao bound will be derived.

$$b(x) = E\left[\hat{x} \mid x\right] - x; \text{ conditional bias}$$

First rule: , (Q-1)

$$b(x) = E\left\{\left[\hat{x}(y) - x\right] \mid x\right\} = \int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] p(y \mid x)\, dy$$

$$\frac{db(x)}{dx} = \frac{dE\left\{\left[\hat{x}(y) - x\right] \mid x\right\}}{dx} = \frac{d\int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] p(y \mid x)\, dy}{dx}$$

$$= \frac{d}{dx}\left\{\int_{-\infty}^{\infty} \hat{x}(y)\, p(y \mid x)\, dy\right\} - \frac{d}{dx}\left\{\int_{-\infty}^{\infty} x p(y \mid x)\, dy\right\}$$

Using Leibniz's Rule:

$$\frac{db(x)}{dx} = \int_{-\infty}^{\infty} \frac{d}{dx}\left[\hat{x}(y)\, p(y \mid x)\right] dy - \int_{-\infty}^{\infty} \frac{d}{dx}\left[x p(y \mid x)\right] dy$$

$$= \int_{-\infty}^{\infty} \hat{x}(y) \frac{dp(y \mid x)}{dx} dy + \int_{-\infty}^{\infty} \frac{d\hat{x}(y)}{dx} p(y \mid x)\, dy - \int_{-\infty}^{\infty} x \frac{dp(y \mid x)}{dx} dy - \int_{-\infty}^{\infty} \frac{dx}{dx} p(y \mid x)\, dy \quad \text{, (Q-2)}$$

$$\frac{db(x)}{dx} = -\int_{-\infty}^{\infty} \frac{dx}{dx} p(y \mid x)\, dy + \int_{-\infty}^{\infty} \hat{x}(y) \frac{dp(y \mid x)}{dx} dy - \int_{-\infty}^{\infty} x \frac{dp(y \mid x)}{dx} dy$$

$$- \int_{-\infty}^{\infty} \frac{dx}{dx} p(y \mid x)\, dy + \int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] \frac{dp(y \mid x)}{dx} dy$$

Chain Rule:

$$\frac{du}{dx} = u \frac{d \ln u}{dx}; \text{ where } u = p(y \mid x) \text{ and } \frac{du}{dx} = p(y \mid x) \frac{d \ln p(y \mid x)}{dx}$$

$$\frac{db(x)}{dx} = -\int_{-\infty}^{\infty} p(y|x)\,dy + \int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] \frac{d \ln p(y|x)}{dx} \left[p(y|x)\right] dy$$

$$\frac{db(x)}{dx} + \underbrace{\int_{-\infty}^{\infty} p(y|x)\,dy}_{=1} = \int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] \frac{d \ln p(y|x)}{dx} \left[p(y|x)\right] dy \quad , \quad \text{(Q-3)}$$

$$\int_{-\infty}^{\infty} \left[\hat{x}(y) - x\right] \frac{d \ln p(y|x)}{dx} \left[p(y|x)\right] dy = 1 + \frac{db(x)}{dx}$$

Shwarz inequality:

$$\left| \int_{-\infty}^{\infty} f^*(t)\, g(t)\, dt \right| \le \sqrt{\int_{-\infty}^{\infty} |f(t)|^2\, dt \int_{-\infty}^{\infty} |g(t)|^2\, dt}$$

$$f(y) = \sqrt{p(y|x)} \left[\hat{x}(y) - x\right] \quad , \text{(Q-4)}$$

$$g(y) = \sqrt{p(y|x)}\, \frac{d \ln p(y|x)}{dx}$$

After substitution equation (Q-3) into equation (Q-4):

$$\left| 1 + \frac{db(x)}{dx} \right| \le \sqrt{\int_{-\infty}^{\infty} \left| \sqrt{p(y|x)} \left[\hat{x}(y) - x\right] \right|^2 dx \int_{-\infty}^{\infty} \left| \sqrt{p(y|x)}\, \frac{d \ln p(y|x)}{dx} \right|^2 dx}$$

$$\le \sqrt{\int_{-\infty}^{\infty} \left| \left[\hat{x}(y) - x\right] \right|^2 p(y|x)\,dx \int_{-\infty}^{\infty} \left| \frac{d \ln p(y|x)}{dx} \right|^2 p(y|x)\,dx} \quad , \quad \text{(Q-5)}$$

$$\le \sqrt{\int_{-\infty}^{\infty} \underbrace{\left| \left[\hat{x}(y) - x\right] \right|^2 p(y|x)\,dx}_{\text{second moment given x}} \int_{-\infty}^{\infty} \underbrace{\left| \frac{d \ln p(y|x)}{dx} \right|^2 p(y|x)\,dx}_{\text{second moment given x}}}$$

Rewriting:

$$\left|1+\frac{db(x)}{dx}\right| \le \left\{E\left[\left(\hat{x}(y)-x\right)^2 \mid x\right]\right\}^{\frac{1}{2}} \left\{E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2 \mid x\right]\right\}^{\frac{1}{2}}$$

$$\left\{E\left[\left(\hat{x}(y)-x\right)^2 \mid x\right]\right\}^{\frac{1}{2}} \ge \frac{\left|1+\frac{db(x)}{dx}\right|}{\left\{E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2 \mid x\right]\right\}^{\frac{1}{2}}}$$

$$E\left[\left(\hat{x}(y)-x\right)^2 \mid x\right] \ge \frac{\left|1+\frac{db(x)}{dx}\right|^2}{E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2 \mid x\right]} \quad , \qquad \text{(Q-6)}$$

For an unconditional bias estsimator, which is used in this research

$$b = E\left[\hat{x}-x\right] = 0$$

And for a known density the condition on x is removed. Then the above reduces to the following:

$$E\left[\left(\hat{x}(y)-x\right)^2\right] \ge \frac{1}{E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2\right]}, \qquad \text{(Q-7)}$$

As per Whalen, the equality in the Schwarz inequality holds only when

$$f(t) = kg(t)$$

$$E\left[\left(\hat{x}(y)-x\right)^2\right] = \frac{1}{E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2\right]}$$

$$\frac{d\ln p(y\mid x)}{dx} = k(x)\left[\hat{x}(y)-x\right] \quad , \qquad \text{(Q-8)}$$

$$E\left[\left(\frac{d\ln p(y\mid x)}{dx}\right)^2 \mid x\right] = \int_{-\infty}^{\infty}\left(\frac{dp(y\mid x)}{dx}\right)^2 p^{-1}(y\mid x)dy$$

Continuing with example from Whalen, using chain rule equation (Q-8) can be simplified to the familiar form of equation (Q-9)

$$u = p(y \mid x); \qquad \frac{du}{dx} = \frac{ud \ln u}{dx}$$

$$\frac{d \ln u}{dx} = \frac{u^{-1} du}{dx}$$

$$E\left[\left(\frac{d}{dx}\right)^2 \ln p(y \mid x)\right] = \int_{-\infty}^{\infty} \left\{\frac{d^2 p(y \mid x)}{dx^2} - \left(\frac{dp(y \mid x)}{dx}\right)^2 p^{-1}(y \mid x)\right\} dy$$

$$= \underbrace{\left(\frac{d}{dx}\right)^2 \underbrace{\int_{-\infty}^{\infty} p(y \mid x) dy}_{=1}}_{=0} - \int_{-\infty}^{\infty} \left(\frac{dp(y \mid x)}{dx}\right)^2 p^{-1}(y \mid x) dy$$

$$E\left[\left(\frac{d \ln p(y \mid x)}{dx}\right)^2 \mid x\right] = -E\left[\left(\frac{d \ln p(y \mid x)}{dx}\right)^2\right]$$

$$E\left\{\left[\hat{x}(y) - x\right]^2\right\} \geq \frac{-1}{E\left[\frac{d^2 \ln p(y \mid x)}{dx^2}\right]}, \qquad \text{(Q-9)}$$

Now to use this information to estimate the Cramér-Rao bound to estmate an unknown parameter.

$$r(t) = s_r(t \mid \underline{x}) + n(t); \qquad t_0 \leq t \leq t_0 + T, \qquad \text{(Q-10)}$$

It is assumed that the signal is known over the same time interval. The probability density function of the received signal in white Gaussian noise is as follows:

$$p(\underline{r}) = \prod_{i=0}^{N_{samp}-1} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(r_i - \tilde{r}_i)^2}{2\sigma_i^2}}; \qquad N_{samp} = \frac{T}{\underbrace{\delta t}_{\text{sampling interval}}}, \qquad \text{(Q-11)}$$

$$\sigma_i^2 = \frac{N_0}{2}$$

$$p\left(\underline{r}\right)=\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N_{samp}} e^{-\left[\sum_{i=0}^{N_{samp}-1}\frac{1}{2\sigma^2}\left(r_i-\tilde{r}_i\right)^2\delta t\right]}$$

$$p\left(\underline{r}\right)=\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N_{samp}}{2}} e^{-\left[\sum_{i=0}^{N_{samp}-1}\frac{1}{2\sigma^2}\left(r_i-\tilde{r}_i\right)^2\delta t\right]}$$

$$=\left(\frac{1}{2\pi\frac{N_0}{2}}\right)^{\frac{N_{samp}}{2}} e^{-\left[\sum_{i=0}^{N_{samp}-1}\frac{1}{2\frac{N_0}{2}}\left(r_i-\tilde{r}_i\right)^2\delta t\right]} \quad , \qquad \text{(Q-12)}$$

$$=\left(\frac{1}{\pi N_0}\right)^{\frac{N_{samp}}{2}} e^{-\left[\sum_{i=0}^{N_{samp}-1}\frac{\left(r_i-\tilde{r}_i\right)^2\delta t}{N_0}\right]}$$

$\overline{r}\left(t\right)=E\left[r\left(t\right)\right];$ Let $B\Rightarrow\infty$, so that $\delta t\Rightarrow 0$

$$\therefore\ p\left(\underline{r}\right)=A\exp\left[-\int_0^T\frac{\left(r_i-\tilde{r}_i\right)^2}{N_0}dt\right] \quad , \qquad \text{(Q-13)}$$

$$R_n\left(\tau\right)=\left(\frac{N_0}{2}\right)\delta\left(\tau\right)$$

$$E\left[r\left(t\right)\right]=s_r\left(t\mid\underline{x}\right);$$

$$\ln p\left(r\mid\underline{x}\right)=\ln\left\{A\exp\left[-\int_0^T\frac{\left(r_i-\tilde{r}_i\right)^2}{N_0}dt\right]\right\}$$

$$\ln p\left(r\mid\underline{x}\right)=\ln A+\ln\left\{\exp\left[-\int_0^T\frac{\left(r_i-\tilde{r}_i\right)^2}{N_0}dt\right]\right\}$$

$$\frac{\partial\ln p\left(r\mid\underline{x}\right)}{\partial x_i}=0+\frac{\partial\left[-\int_0^T\frac{\left(r_i-\tilde{r}_i\right)^2}{N_0}dt\right]}{\partial x_i}$$

Substitute equation (Q-10) into equation (Q-13):

$$\frac{\partial \ln p\left(r \mid \underline{x}\right)}{\partial x_i} = \frac{\partial\left[-\int_0^T \frac{\left(r_i - \tilde{r}_i\right)^2}{N_0} dt\right]}{\partial x_i}; \text{ substitute in for the received signals} \quad \text{(Q-14)}$$

$$= -\frac{1}{N_0}\frac{\partial}{\partial x_i}\left[\int_0^T \left(r(t) - s_r\left(t \mid \underline{x}\right)\right)^2 dt\right]$$

Now taking the partial derivatives

$$\frac{\partial \ln p\left(r \mid \underline{x}\right)}{\partial x_i} = -\frac{1}{N_0}\int_0^T 2\left(r(t) - s_r\left(t \mid \underline{x}\right)\right)^2 \frac{\partial}{\partial x_i}\left[s_r\left(t \mid \underline{x}\right)\right] dt$$

$$r(t) = s_r\left(t \mid x\right) - n(t) \quad \text{(Q-15)}$$

$$r(t) - s_r\left(t \mid x\right) = n(t); \text{ where}$$

$$E\left[n(t)n^*(t)\right] = R_n\left(t - t'\right) = \left(\frac{N_0}{2}\right)\delta\left(t - t'\right)$$

The noiseless complex envelop of the received signal:

$$s_r(t) = Aa(t)\cos\left(\omega_c t + \phi(t) + \theta_c\right)$$

$$\tilde{r}(t) = A\tilde{s}_{r_0}(t)\exp\left(j\theta_c\right) + \tilde{n}(t); \text{ where}, \quad \text{(Q-16)}$$

$$\tilde{s}_{r_0}(t) = a(t)\exp\left[j\phi(t)\right]$$

As per Whalen, θ_c is considered a nuisance parameter of Uniform Density and will be removed by taking the expectation given θ_c.

$$E\left[\tilde{r}(t) \mid \theta_c\right] = \int_{-\pi}^{\pi} A\exp\left[-\int_0^T \frac{\left(\tilde{r} - \tilde{s}_r(t)\right)^2}{N_0} dt\right] p_\theta(\theta) d\theta; \text{ where } p_\theta(\theta) = \frac{1}{2\pi}$$

$$= \int_{-\pi}^{\pi} A\exp\left[-\int_0^T \frac{\left(\tilde{r} - \tilde{s}_r(t)\right)^2}{N_0} dt\right]\frac{1}{2\pi} d\theta \quad \text{(Q-17)}$$

$$= \frac{A}{2\pi}\int_{-\pi}^{\pi} \exp\left[-\int_0^T \frac{\left(\tilde{r} - \tilde{s}_r(t)\right)^2}{N_0} dt\right] d\theta$$

Now substitute equation (Q-16) into equation (Q-17) as follows:

$$E\left[\tilde{r}(t)\,|\,\theta_c\right]=\frac{A}{2\pi}\int_{-\pi}^{\pi}\exp\left[-\int_0^T\frac{\left(\tilde{r}-\tilde{s}_r(t)\right)^2}{N_0}dt\right]d\theta\text{; where}$$

$$\tilde{s}(t)=Aa(t)\exp\left\{j\left[\phi(t)+\theta_c\right]\right\}\text{ and }\frac{1}{T}\int_0^T a^2(t)\,dt=1$$

$$E\left[\tilde{r}(t)\,|\,\theta_c\right]=\frac{A}{2\pi}\int_{-\pi}^{\pi}\exp\left[-\frac{1}{N_0}\int_0^T\left(\underbrace{\tilde{r}^2(t)}_{\tilde{r}(t)\tilde{r}^*(t)}-2\tilde{r}(t)\tilde{s}_r(t)+\underbrace{\tilde{s}^2(t)}_{\tilde{s}(t)\tilde{s}^*(t)}\right)dt\right]d\theta$$

$$=\frac{A}{2\pi}\int_{-\pi}^{\pi}\exp\left[-\frac{1}{N_0}\int_0^T\left(\left|\tilde{r}(t)\right|^2-2Aa(t)\tilde{r}(t)\exp\left\{j\left[\phi(t)+\theta\right]\right\}+\left[|A|^2\left|a(t)\right|^2\underbrace{e^{j(\phi(t)+\theta)}e^{-j(\phi(t)+\theta)}}_{=1}\right]\right)dt\right]d\theta$$

$$=\frac{A}{2\pi}\left\{\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+|A|^2\left|a(t)\right|^2\right]dt\right]\int_{-\pi}^{\pi}\exp\left[\frac{1}{N_0}\int_0^T\left(2Aa(t)\tilde{r}(t)\exp\left\{j\left[\phi(t)+\theta\right]\right\}\right)dt\right]d\theta\right\}$$

, (Q-18)

For convience Whalen redefines the last integral term as follows:

$$\int_0^T a(t)\tilde{r}(t)\exp\left[j\phi(t)\right]dt=2q\exp(j\beta)$$

$\therefore$

$$E\left[\tilde{r}(t)\,|\,\theta\right]=\frac{A}{2\pi}\left\{\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+|A|^2\left|a(t)\right|^2\right]dt\right]\int_{-\pi}^{\pi}\exp\left[\int_0^T\frac{\left[2Aq\exp(j\beta)\exp(j\theta)\right]}{N_0}dt\right]d\theta\right\}$$

, (Q-19)

Where the modified Bessel function is defined as follows:

$$2\int_0^{\pi}\exp\left[\gamma\cos(\theta)\right]d\theta=2\pi I_0(\gamma)$$

$$E\left[\tilde{r}(t)\,|\,\theta\right]=\frac{A}{2\pi}\left\{\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+|A|^2\left|a(t)\right|^2\right]dt\right]\int_{-\pi}^{\pi}\exp\left[\int_0^T\frac{\left[2Aq\exp(j\beta)\exp(j\theta)\right]}{N_0}dt\right]d\theta\right\}$$

, (Q-20)

$$E\left[\tilde{r}(t)\,|\,\theta\right] = 2\pi\frac{A}{2\pi}I_0\left(\frac{2Aq}{N_0}\right)\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+\left|A\right|^2\left|a(t)\right|^2\right]dt\right]$$

dropping the given θ term

$$E\left[\tilde{r}(t)\right] = AI_0\left(\frac{2Aq}{N_0}\right)\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+\left|A\right|^2\left|a(t)\right|^2\right]dt\right] \quad , \quad \text{(Q-21)}$$

$$\ln E\left[\tilde{r}(t)\right] \cong \ln\left\{AI_0\left(\frac{2Aq}{N_0}\right)\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+\left|A\right|^2\left|a(t)\right|^2\right]dt\right]\right\}$$

remembering that $\left|a(t)\right|^2 = \left|\tilde{s}_{r_0}(t)\right|^2$

$$\ln E\left[\tilde{r}(t)\right] \cong \ln A + \ln\left[I_0\left(\frac{2Aq}{N_0}\right)\right] + \ln\left\{\exp\left[-\frac{1}{N_0}\int_0^T\left[\left|\tilde{r}(t)\right|^2+\left|A\right|^2\left|\tilde{s}_{r_0}(t)\right|^2\right]dt\right]\right\}$$

for high SNR: $\tilde{r}(t) = A\tilde{s}_{r_0}(t)e^{j\theta}$, as defined earlier less the noise term

for a large argument the natural log of the modified Bessel function is as follows:

$\ln I_0(x) \cong x$

$\therefore$

$$\ln E\left[\tilde{r}(t)\right] \cong \ln A + \frac{2Aq}{N_0} + \left[-\frac{1}{N_0}\int_0^T\left[\left|A\right|^2\tilde{s}_{r_0}(t)\tilde{s}_{r_0}^*(t)e^{j\theta}e^{-j\theta}+\left|A\right|^2\left|\tilde{s}_{r_0}(t)\right|^2\right]dt\right] \quad , \quad \text{(Q-22)}$$

$$= \ln A + \frac{2Aq}{N_0} + \left[-\frac{1}{N_0}\int_0^T\left[\left|A\right|^2\left|\tilde{s}_{r_0}(t)\right|^2+\left|A\right|^2\left|\tilde{s}_{r_0}(t)\right|^2\right]dt\right]$$

$$= \ln A + \frac{2Aq}{N_0} - \frac{2A^2}{N_0}\int_0^T\left[\left|\tilde{s}_{r_0}(t)\right|^2\right]dt$$

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. US Special Operations Command
   Macdill Air Force Base
   Tampa, FL

4. Naval Information Operations Center, Suitland
   Office of Naval Intelligence
   Suitland, MD

5. Martin Kruger
   Office of Naval Research
   Washington, D. C.

6. George Hinckley
   VARDI
   Quantico, VA

7. Prof. John McEachen
   Naval Postgraduate School
   Monterey, CA

8. Prof. Murali Tummala
   Naval Postgraduate School
   Monterey, CA

9. Prof. Frank Kragh
   Naval Postgraduate School
   Monterey, CA

10. Prof. Wei Su
    Naval Postgraduate School
    Monterey, CA

11. Chris Eagle
    Naval Postgraduate School
    Monterey, CA